

Comparative Study of Advanced Turbulence Models for Turbomachinery

Ali H. Hadid and Munir M. Sindir

**Rocketdyne Division
Boeing North American Inc.
CFD Technology Center
Canoga Park, CA 91309**

**Contract NAS8-38860
Final Report**

Approved by



**Glenn Havskjold
Program Manager**

October 1996

Comparative Study of Advanced Turbulence Models for Turbomachinery

TABLE OF CONTENTS

	<u>Page</u>
ACKNOWLEDGMENTS	4
SUMMARY	5
1. INTRODUCTION	
1.1 Background	6
1.2 Outline of Present Study	7
2. 2D/Axisymmetric k-ε Model	9
2.1 Introduction	10
2.2 Theory and Model Equations	10
2.3 Module Evaluation	15
References	18
Figures	
Appendix A	20
3. 2D/Axisymmetric Multi-Scale k-ε Model	29
3.1 Introduction	30
3.2 Theory and Model Equations	30
3.3 Module Evaluation	34
References	36
Figures	
Appendix B	37
4. 2D/Axisymmetric Algebraic Stress Model	48
4.1 Introduction	49
4.2 Theory and Model Equations	49

4.3	Module Evaluation	52
	References	53
	Figures	
	Appendix C	54
5.	2D/Axisymmetric Reynolds Stress Model	58
5.1	Introduction	59
5.2	Theory and Model Equations	59
5.3	Boundary Conditions	67
5.4	Numerical Procedure	68
5.4.1	Wall Reflection Treatment	71
5.5	Module Evaluation	72
	References	74
	Figures	
	Appendix D	76
6.	3D k - ε Turbulence Model	80
6.1	Introduction	81
6.2	Theory and Model Equations	81
6.3	Module Evaluation	82
	References	82
	Figures	
	Appendix E	83
7.	3D Algebraic Stress Model	90
7.1	Introduction	91
7.2	Theory and Model Equations	91
	References	94
	Appendix F	95
8.	Related Publications and presentations	98

ACKNOWLEDGEMENTS

This work was supported jointly by NASA Marshall Space Flight Center under contract NAS8-38860 and by Rocketdyne Division of Rockwell International under discretionary funds. The authors gratefully acknowledge the efforts of NASA contract monitors Dr. Paul McConnaughey and Mr. Robert Garcia and Rocketdyn program manager Dr. Glenn Havskjold. The authors would also like to acknowledge the efforts of Professor C.-P. Chen and his colleagues Dr. Hong Wei and Ms. X. Li at the University of Alabama at Huntsville for their help in testing the modules. The assistance of Mr. Steve Barson in managing the project and the efforts of Ms. Michelle DeCroix and Mr. Armen Darian of Rocketdyne in testing some of the modules are greatly appreciated.

SUMMARY

A computational study has been undertaken to study the performance of advanced phenomenological turbulence models coded in a modular form to describe incompressible turbulent flow behavior in two dimensional/axisymmetric and three dimensional complex geometry. The models include a variety of two equation models (single and multi-scale $k-\epsilon$ models with different near wall treatments) and second moment algebraic and full Reynolds stress closure models. These models were systematically assessed to evaluate their performance in complex flows with rotation, curvature and separation. The models are coded as self contained modules that can be interfaced with a number of flow solvers. These modules are stand alone satellite programs that come with their own formulation, finite-volume discretization scheme, solver and boundary condition implementation. They will take as input (from any generic Navier-Stokes solver) the velocity field, grid (structured H-type grid) and computational domain specification (boundary conditions), and will deliver, depending on the model used, turbulent viscosity, or the components of the Reynolds stress tensor $\overline{u_i u_j}$. There are separate 2D/axisymmetric and/or 3D decks for each module considered.

The modules are tested using Rocketdyn's proprietary code REACT. The code utilizes an efficient solution procedure to solve Navier-Stokes equations in a non-orthogonal body-fitted coordinate system. The differential equations are discretized over a finite-volume grid using a non-staggered variable arrangement and an efficient solution procedure based on the SIMPLE algorithm for the velocity-pressure coupling is used. The modules developed have been interfaced and tested using finite-volume, pressure-correction CFD solvers which are widely used in the CFD community. Other solvers can also be used to test these modules since they are independently structured with their own discretization scheme and solver methodology. Many of these modules have been independently tested by Professor C.P. Chen and his group at the University of Alabama at Huntsville (UAH) by interfacing them with own flow solver (MAST).

CHAPTER 1

Introduction

1.1 Background

Computational Fluid Dynamics (CFD) has been used extensively for the last decade or so in analyzing complex flow phenomenon for many industrial applications, such as combustion and turbomachinery. Most flows of practical interest are turbulent and for many of them, relatively simple prediction methods are sufficient to produce results of engineering accuracy. For others, mainly flows in complex geometry with large body forces such as curvature, rotation and separation, more complex prediction methods are required.

With advancing state-of-the-art of computer technology, the range, size and complexity of flow models being applied have increased. Users have become more sophisticated and there is a constant demand for improvement. CFD codes have adapted to this demand and many general-purpose computer codes have been developed and used. As these general purpose codes become larger, their code structure becomes sophisticated and in general this structure can be divided into three main areas;

- 1) Numerical algorithms which include discretization methods and solution techniques.
- 2) Methods of dealing with complex geometry, such as grid generation, structured or unstructured grids.
- 3) Physical models which include turbulence models, porosity, combustion kinetics, multi-phase flows, etc.

It seems, therefore, that the practicing engineer must have the knowledge of all these elements of the CFD program in order to successfully utilize the code. Modularization of the code structure may then become necessary in order to obtain the maximum benefits from these general-purpose CFD codes. This means developing individual modular routines for the solver and other physical models. If such modules are successful they would allow users to concentrate their talents on developing and improving physical hypothesis such as turbulence models that can be easily tested using these modules.

In general, the physics of turbulence can be captured by solving the full time-dependent Navier-Stokes equations in what is termed as Direct Numerical Simulation (DNS). However, DNS is not practical for engineering purposes mainly because it is restricted to flows at low Reynolds numbers. Large Eddy Simulations (LES) are now competitive with DNS in accuracy at an order of magnitude less cost, however, it is still expensive for routine engineering calculations. Therefore, current engineering prediction methods are based on Reynolds-averaged equations, with models for the unknown Reynolds stresses which appear as the result of time-averaging the nonlinear Navier-Stokes equations. These models fall mainly into three categories; "eddy-viscosity" models, where a relation between the Reynolds stresses and mean velocity gradients at the same point in space is sought. Algebraic stress models, where the Reynolds stresses are expressed as an algebraic relation of turbulence production and dissipation. Reynolds stress models where the exact partial differential equations for the Reynolds stresses are solved after closing the higher order terms. These transport equations account for the dependence of Reynolds stresses on the history of the flow and should perform better than the eddy-viscosity models.

1.2 Outline of the Present Study

In the present work, phenomenological, single-point turbulence models coded in a modular format are developed as self-contained code decks that can be interfaced with a number of flow solvers to analyze turbulent flows in complex 2D/axisymmetric or 3D geometry. These modules are validated using Rocketdyn's REACT code and are independently tested at UAH using own code MAST.

The models that are developed in a modular form include;

1. 2D/axisymmetric single-scale $k-\varepsilon$ model with three options for near wall treatment that include;
 - Standard Launder and Spalding wall functions.
 - Chen and Patel two-layer model.
 - Lam and Bremhorst low-Reynolds number model.
2. 2D/axisymmetric multi-scale $k-\varepsilon$ model with the standard wall function and Chen & Patel two-layer near wall treatment.
3. 2D/axisymmetric implicit algebraic stress model (ASM) based on the original work of Rodi.
4. 2D/axisymmetric full Reynolds stress turbulence model (RSM) based on the simplified linear

second moment closure model of Launder, Reece and Rodi (LRR) second moment closure.

5. 3D standard $k-\epsilon$ turbulence model with wall function and two-layer near wall treatments.
6. 3D algebraic stress model (ASM).

Each model is coded as a self contained, stand alone module deck that can be interfaced with a number of CFD solvers to analyze turbulent flows in complex geometry. The user can use these modules without concern as to how they are implemented and solved. The input to the modules are the mean flow variables, boundary and geometric information which are to be provided by a mean flow solver. The output of the module are the turbulent eddy-viscosity for the eddy-viscosity models and the Reynolds stresses for the second moment closure models. Moreover, source terms which are needed for the mean flow calculations are calculated and must be passed to the main solver. The modules are tested using the finite-volume REACT code and the results compared with available experimental data.

Full details of each module are given in the next chapters. Chapter 2 discusses the theory and model equations for the two-equation $k-\epsilon$ model used in the 2D/axisymmetric module deck. The module is evaluated with a number of benchmark problems and detailed description of the module variable names together with the input/output structure are given in appendix A. The complete listing of the module is provided at the end of the chapter. Similarly, chapter 3 discusses the theory and model equations for the 2D/axisymmetric multi-time-scale $k-\epsilon$ model. The 2D/axisymmetric Algebraic stress module is presented in chapter 4 and chapter 5 discusses the 2D/axisymmetric Reynolds stress module deck. Full description of the 3D $k-\epsilon$ turbulence model is given in chapter 6 and chapter 7 presents a full description of the 3D algebraic stress model together with module description and code listing in the appendix. Finally in chapter 8, copies of related turbulence work that are presented or published elsewhere are attached.

CHAPTER 2

2D/Axisymmetric k - ε Turbulence Model

Table of Contents

	page
2.1 Introduction	10
2.2 Theory and Model Equations	10
2.3 Module Evaluation	15
References	18
Figures	
Appendix A	20

2.1 Introduction

In this section a description of the standard k - ϵ turbulence model that is coded as a self contained computer program to compute turbulent flow quantities in two-dimensional planar or axisymmetric geometry is given. Detailed description of the module structure, variables used and how to interface the module with CFD flow solvers are given in Appendix A. The module has been tested as a separate self-contained unit using the REACT code [1] and was independently tested at the University of Alabama at Huntsville (UAH) using own code (MAST).

2.2 Theory and Model Equations

The k - ϵ turbulence module is based on the widely used single-scale two equation k - ϵ turbulence model (k is the turbulent kinetic energy and ϵ is the energy dissipation rate). The model developed originally by Launder and Spalding [2] was successful in providing good predictions for a wide range of turbulent flows. The k and ϵ -equations can be derived from the transport equations for the Reynolds stresses assuming fully turbulent flow.

For low-Reynolds number flows close to solid boundaries, adjustments to the model are needed to bridge the viscous dominated sublayer region with the fully turbulent flow region. The success of the wall function method depends on the universality of the turbulent flow structure near the wall. In many complex flows, however, the flow field near the wall has to be determined accurately and the traditional wall-function method is not satisfactory. This is because the specification of all turbulence quantities in terms of the friction velocity fail at separation where the flow near the wall is no longer controlled by the wall shear stress. Patel et al [3] assessed the relative performance of various models which describe the near-wall flows and found that there are still areas of improvements needed to accurately model flow behavior near the wall.

Jones and Launder [4] extended the original k - ϵ model to the low-Reynolds number form which allowed the calculation to be performed all the way to the wall. Numerical difficulties of accurately resolving the large gradients close to the wall necessitates resolving the wall region with a very fine grid structure. Chen and Patel [5] introduced a method to resolve the near-wall region which combines the standard k - ϵ model with the one-equation model of Wolfshtein [6] near the wall. In this "two-layer" model an algebraically prescribed eddy-viscosity for the wall region is coupled to the k - ϵ model to describe the details of the flow in the vicinity of the wall.

Momentum and continuity equations are solved up to the wall and this reduces the physical uncertainties of near-wall turbulence and the numerical difficulties of resolving the very large gradients of turbulence parameters.

For an incompressible, steady and axisymmetric turbulent flow, the Reynolds averaged momentum and continuity equations can be expressed in a generalized form as;

$$\frac{\partial(\rho u \Phi)}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r} (\rho v r \Phi) = \frac{\partial}{\partial x} (\Gamma \Phi_x \frac{\partial \Phi}{\partial x}) + \frac{1}{r} \frac{\partial}{\partial r} (r \Gamma \Phi_r \frac{\partial \Phi}{\partial r}) + S_\Phi \quad (1)$$

where Φ is the dependent variable, which stands for $\Phi = u, v, w$ for the axial, radial and tangential velocities respectively. ρ is the fluid density, $\Gamma \Phi_x$ and $\Gamma \Phi_r$ are exchange coefficients in x and r -directions, respectively, and S_Φ is the source term for the variable Φ .

The source terms for the dependent variable are:

- Axial direction, $\Phi = u$, $\Gamma \Phi_x = 2\mu_e$, $\Gamma \Phi_r = \mu_e$ and

$$S_u = -\frac{\partial P}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r} (\mu_e r \frac{\partial v}{\partial x}) \quad (2)$$

where μ_e is the eddy viscosity and P is the pressure

- Radial direction, $\Phi = v$, $\Gamma \Phi_x = \mu_e$, $\Gamma \Phi_r = 2\mu_e$ and

$$S_v = -\frac{\partial}{\partial x} \left(\mu_e \frac{\partial u}{\partial r} \right) - 2\mu_e \frac{v}{r^2} + \frac{\rho w^2}{r} - \frac{\partial P}{\partial r} \quad (3)$$

- Tangential direction, $\Phi = w$, $\Gamma \Phi_x = \mu_e$, $\Gamma \Phi_r = \mu_e$ and

$$S_w = -\frac{\rho v w}{r} - \frac{w}{r^2} \frac{\partial}{\partial r} (r \mu_e) \quad (4)$$

Equations 2, 3, and 4 above are the momentum equations that are solved by the CFD solvers. However, in order to close the equations and determine the eddy viscosity different turbulence models are used.

The present module utilizes the k - ε model. In this model two equations for the turbulent kinetic energy k and its dissipation ε which have the same general form as equation (1) are solved.

For the turbulent kinetic energy equation

$$\Phi = k, \quad \Gamma_{\Phi_x} = \Gamma_{\Phi_r} = \mu + \frac{\mu_t}{\sigma_k} \quad \text{and} \quad S_{\Phi} = G - \rho\varepsilon \quad (5)$$

For the energy dissipation equation

$$\Phi = \varepsilon, \quad \Gamma_{\Phi_x} = \Gamma_{\Phi_r} = \mu + \frac{\mu_t}{\sigma_{\varepsilon}} \quad \text{and} \quad S_{\Phi} = \frac{\varepsilon}{k} (C_1 f_1 G - C_2 f_2 \rho\varepsilon) \quad (6)$$

where σ_k and σ_{ε} are turbulent Prandtl/Schmidt numbers for k and ε respectively, and G denotes the rate of production of the turbulent kinetic energy and is expressed as:

$$G = \mu_e \left\{ 2 \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial r} \right)^2 + \left(\frac{v}{r} \right)^2 \right] + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial w}{\partial r} - \frac{w}{r} \right)^2 \right\} \quad (7)$$

where μ is the dynamic viscosity, and μ_t is the turbulent viscosity,

$$\mu_t = C_{\mu} f_{\mu} \rho \frac{k^2}{\varepsilon} \quad (8)$$

and $\mu_e = \mu + \mu_t$

C_{μ} , C_1 , C_2 , σ_k and σ_{ε} are constants whose values are 0.09, 1.44, 1.92, 1.0, 1.0, respectively and f_1 , f_2 and f_m are damping functions.

Near a wall, turbulent flow can be divided into two regions, the inner viscous sublayer where low turbulent Reynolds number effects are important and the velocities decrease rapidly to zero at the wall, and the outer fully turbulent region. The successful application of the k - ε turbulence model for many complex flows depends to a large extent on how accurately the flow field near the wall is

determined. In the present module three different models are used to treat this thin sublayer region, they include;

Wall function method, where

$$u^+ = y^+ \quad \text{at } y^+ \leq 11.6 \quad (9)$$

$$u^+ = \frac{1}{\kappa} \ln(E y^+) \quad \text{at } y^+ \geq 11.6 \quad (10)$$

where, $u^+ = \frac{u}{u_\tau}$, $y^+ = \frac{u_\tau y}{\nu}$ and $u_\tau = \sqrt{\tau_w / \rho}$

τ_w is the wall shear stress which can be determined from

$$\tau_w = \frac{\mu u_p}{\delta} \quad \text{for } y^+ \leq 11.6 \quad (11)$$

$$\tau_w = \frac{\kappa C_\mu^{0.25} \rho u_p k^{0.5}}{\ln [E C_\mu^{0.25} \rho \delta k^{0.5} / \mu]} \quad \text{for } y^+ > 11.6 \quad (12)$$

Here, u_p denotes the velocity component parallel to the wall at the first grid point p from the wall.

δ is the normal distance from the wall and κ is a constant = 0.42.

In this approach, k and ε equations are solved with $f_\mu = f_1 = f_2 = 1$, only in the fully turbulent region beyond some distance from the wall. Boundary conditions i.e., velocity components and turbulent parameters at that distance are specified in terms of the friction velocity u_τ .

In the low-Reynolds number model, the flow is resolved all the way to the wall with a very fine mesh. Many models have been proposed that are based on the k - ε model and differ mainly in the choice of the damping functions f_μ , f_1 and f_2 to bridge the gap between the sublayer and the fully turbulent region. The model due to Lam & Bremhorst [7] is used in this work, where;

$$f_\mu = [1 - \exp(-0.016 R_y)]^{1/2} (1 + \frac{20.5}{R_t})$$

$$f_1 = 1 + \left(\frac{0.06}{f_\mu} \right)^3 \text{ and } f_2 = 1 - \exp(-R_t^2)$$

where, $R_y = \frac{k^{1/2}y}{\nu}$ and $R_t = \frac{k^2}{\nu \epsilon}$ are turbulent Reynolds number.

These damping functions tend to unity with increasing distance from the wall.

In the two-layer model due to Chen and Patel [5], a simple algebraically prescribed eddy-viscosity model for the wall region is coupled to the k - ϵ model for the outer flow to describe the flow details. Unlike the low-Reynolds number model that requires the solution of transport equations for both k and ϵ all the way to the wall, the one-equation model requires the solution of only the turbulent kinetic energy equation in the sublayer region while algebraically specifying the eddy viscosity and energy dissipation.

$$\nu_t = C_\mu \frac{k^{1/2}}{L_\mu} \text{ and } \epsilon = \frac{k^{3/2}}{L_\epsilon}.$$

The length scales L_μ and L_ϵ contain the necessary damping effects in the near-wall region in terms of the turbulence Reynolds number R_y .

$$L_\mu = C_l y [1 - \exp(-R_y/A_\mu)] \quad (13)$$

$$L_\epsilon = C_l y [1 - \exp(-R_y/A_\epsilon)] \quad (14)$$

L_μ and L_ϵ become linear and approach $C_l y$ with increasing distance from the wall.

$C_l = \kappa C_\mu^{-0.75}$ and $A_\epsilon = 2C_l$. Chen and Patel [5] used $A_\mu = 70$.

The damping effects decay rapidly with distance from the wall independent of the magnitude of the wall shear stress. The matching between the one-equation and the standard k - ϵ models is carried along prescribed grid lines where $R_y \sim 200$.

For flows in rotating ducts a modification was made by Chen and Guo [8] to reflect the effects of a system rotation on the length scales L_μ and L_ϵ , as;

$$L_{\mu} = L_{\mu 0} [1.0 + 1.3 (0.4 \frac{\partial U}{\partial y} - 0.8 \Omega) \Omega (\frac{k}{\varepsilon})^2]^{1.5}$$

$$L_{\varepsilon} = L_{\varepsilon 0} [1.0 + 1.3 (0.4 \frac{\partial U}{\partial y} - 0.8 \Omega) \Omega (\frac{k}{\varepsilon})^2]^{0.5}$$

Moreover, the function f_2 in the dissipation equation is modified to

$$f_2 = f_2 + Ri$$

where Ri is a Richardson number to reflect the effects of streamline curvature due to rotation and is defined as

$$Ri = (0.4 \omega_k - 0.8 \Omega_k) \Omega_k (\frac{k}{\varepsilon})^2$$

where $\omega_k = \epsilon_{ijk} \frac{\partial U_i}{\partial x_j}$ is the local mean vorticity.

The above modification to account for streamline curvature and rotation seemed adequate in the framework of two equation k - ε modeling. Other modifications have also been considered but not implemented in this module and can be referred to in Hadid and Sindir [9].

2.3 Module Evaluation

The single scale k - ε turbulent module was evaluated by comparison with published experimental data. One of the test problems considered is the two dimensional incompressible turbulent flow over a backward facing step with and without rotation (see figure 1) to compare with the experiment of Rothe and Johnston[10]. While the mean flow is in the x - y plane, the channel is rotated with constant angular velocity Ω about the z -axis. The ratio of the channel width to the step height is very large so that the secondary flow can be ignored, which made the flow remain two dimensional. The channel height to step ratio was set to 2 and the inlet channel height (h) equals to the step height (H). The Reynolds number based on the uniform inlet velocity was about 5500. The rotation number ($Ro = \Omega h / U$) was varied between $+0.06$ and -0.06 .

The streamline patterns for the three different rotation numbers $Ro = -0.06, 0.0, +0.06$ by using the three different wall treatments are shown in figures 2-4. In each figure, the upper (a) and lower (c) parts correspond to $Ro = +0.06$ and $Ro = -0.06$ respectively. While the middle part (b) is the non-rotating case. It is observed that the streamline patterns are influenced by the system rotation. Suction side step extends the recirculation zone and the pressure side step reduces the recirculation zone. The reattachment length for $Ro = -0.06$ using the wall functions is larger compared to the

experimental results. This is due to the fact that no Coriolis effect is accounted for in the law of the wall. The predicted variation of reattachment length with Ro (figure 5) shows reasonable correlation with the experimental data of Rothe and Johnston [10].

The single scale $k-\varepsilon$ model using three different wall treatments with rotational stress generation terms embodied seems to capture the main effects of system rotation on turbulence structure, i.e. the suppression of turbulence level with clockwise rotation and enhancement of turbulence level with counterclockwise rotation. The effects are also noticeable in the corresponding increase in the reattachment length with clockwise rotation and its decrease with counterclockwise rotation.

The other two test cases were those of Daily and Nece [11] where rotating disk cavity circulation and secondary flows are induced by a rotating wall, and Roback and Johnson [12] for a confined double concentric jets with a sudden expansion. Flow swirl in this case is induced by imposing a tangential velocity component at the outer jet. Figure (6) shows the two-dimensional axisymmetric rotating lid cavity of Daily and Nece. The flow is bounded by a disk (rotor) and a stationary end wall (stator) of a chamber. The ratio of the axial clearance between the rotor and the stator (s) to the radius of the disk (a) is 0.0255 . The disk rotates with a rotational Reynolds number $R=4.4 \times 10^6$ defined as $R = \Omega a^2/\nu$, where Ω is the disk rotational speed and ν is the kinematic viscosity.

Computations were performed on a 33×75 grid with different grid clustering near the walls for the different near-wall models. Figure (7) shows the velocity vectors at the top region of the cavity using the wall function model. Centrifugal forces move the fluid radially outward on the disk, axially away from the disk on the wall casing, and radially inwards on the stationary end wall. Figure 8, shows the axial variations of the radial velocity component at a radial position $r/a=0.765$. The agreement is fair with some discrepancy for all near-wall models close to the rotating disk. Figure (9), shows the axial variation of the tangential velocity component at the radial position $r/a=0.765$. At the rotating disk ($x=0$), the tangential velocity approaches the value $(a\Omega)$. The two-layer near wall model seem to offer closer agreement with the data than the other two models. The presence of corner regions presents a difficulty in defining the normal distances used in the definition of turbulent Reynolds number. In the present analysis, values of the normal distance were based on the normal distance to the nearest solid boundary.

Predictions of the experiments of Roback and Johnson [12] have been presented by several workers, e.g. Sloan et al. [13] and Durst and Wennerberg [14]. Unfortunately, inlet flow profiles were not provided in the experiment. Therefore, the present calculations were started at the

expansion plane using the measured velocity profile at 5 mm downstream of the expansion after some adjustments near the edges of the coaxial jets. Measurements of main turbulent intensities were used to calculate inlet values of the turbulent kinetic energy. Energy dissipation rate was estimated from $\varepsilon = C_\mu k^{3/2}/L$, where L is a length scale of turbulence at the inlet of the order of 10^{-4} m .

Figure 10, shows an illustration of the test chamber geometry. The chamber diameter is about twice the secondary tube diameter. The exit from the 8-bladed, 30° , free vortex swirl generator is located approximately 0.005 m upstream from the confluence plane.

A prominent phenomenon in axisymmetric swirling flows in such geometry is the "bubble" or vortex breakdown which has been studied extensively [15-18]. In the present numerical simulation of the experiment, a 150×100 grid nodes was used with different clustering on the walls for the different near-wall models used. Figure 11, shows the velocity vectors indicating the presence of a closed recirculation zone at the center with additional zones at the corner downstream and between the inner jet and the outward diverted secondary jet. The figure also shows flow diversion outwards with high gradients characterized by large turbulent shear and fluctuation levels. Comparisons were made of the radial variations of flow variables at two axial locations, $x=0.025\text{m}$ upstream of the vortex bubble and $x=0.102\text{m}$ located inside the bubble. Figure (12), shows the radial variation of the axial velocity profile at $x=0.025\text{m}$ using the wall function, two-layer and the low Reynolds number models. Fair agreement by the different models is shown. They also seem to predict small negative velocities at a radial position $r \sim 0.0153\text{m}$ (the interface between the inner and outer jets), slightly under predicted in strength and width. Figure (13) shows the radial variation of the axial velocity profiles at $x=0.102\text{m}$. The two-layer model shows a better agreement with the experimental data.

Radial variations of the tangential velocity at $x=0.025\text{m}$ is shown in figures 14. The figure shows that the two-layer model offers better agreement with the experiment as compared with the wall function or the low Reynolds number models.

In general, the calculations shown above indicate that the two layer model seem to offer a better comparisons with the experimental results. The three near-wall models are built in the standard two-dimensional/axisymmetric $k-\varepsilon$ turbulence module. The structure of the module will be discussed next together with the details of interfacing with a flow solver and descriptions of variables.

REFERENCES

1. Chon, J., Hadid, A. and Hamakiotes C. "REACT-2D Version 2.6 User's Manual", CFD Technology Center, Rocketdyne Division/Rockwell International, 1989
2. Launder, B. E and Spalding, D. B. "The Numerical Computation of Turbulent Flows", Computer Methods in Applied Mechanics and Engineering, vol. 3, pp. 269-289, 1974.
3. Patel, V. C., Rodi, W. and Scheuerer, G. "Turbulence Models for Near-Wall and Low-Reynolds Number Flows: A Review", AIAA Journal, vol. 23, no. 9, pp. 1308-1319, 1985.
4. Jones, W. P. and Launder, B. E. "The Calculation of Low-Reynolds Number Phenomena With a Two-Equation Model of Turbulence", Int. J. Heat and Mass Transfer, vol. 16, pp. 1119-1130, 1973.
5. Chen, H. C. and Patel, V. C. "Near-Wall Turbulence Models for Complex Flows Including Separation", AIAA Journal, vol. 26, no. 6, pp. 641-648, 1988.
6. Wolfshtein, M. "The Velocity and Temperature Distribution in One-Dimensional Flow With Turbulence Augmentation and Pressure Gradients", Int. J. Heat and Mass Transfer, vol. 12, pp. 301-318, 1969.
7. Lam, C. and Bremhorst, K. "A Modified Form of the $k-\epsilon$ Model for Predicting Wall Turbulence", Trans. ASME J. Fluids Eng., vol. 103, pp. 456-460, 1981.
8. Chen, C. P. and Guo, K. L. " Applications of a Two-Layer Near Wall Model to Fully Developed and Rotation Channel Turbulent Flows", NASA Contract Report, 1991.
9. Hadid, A. H. and Sindir, M. M. "Comparative Study of Advanced Turbulence Models for Turbomachinery" NASA CP-3174, 1992.
10. Rothe, P. H. and Johnston, J. P. "Free Shear Layer Behavior in Rotating Systems" Trans. ASME J. Fluids Eng., vol. 103, pp. 456-460, 1981.
11. Daily, J. W. and Nece, R. E. "Chamber Dimension Effects on Induced Flow and Frictional

Resistance of Enclosed Rotating Disks" Trans. ASME J. Basic Eng., pp. 217-232, 1960.

12. Roback, R. and Johnson, B. "Mass and Momentum Turbulent Transport Experiment With Confined Swirling Co-Axial Jets: NASA CR-168252, 1983.
13. Sloan, D. G., Smith, P. J. and Smoot, L. D. "Modeling Swirl in Turbulent Flow Systems" Prog. Energy Combust. Sci., vol. 12, pp. 163-250, 1986.
14. Durst, F. and Wennerberg, D. "Numerical Aspects of Calculations of Confined Swirling Flows With Internal Recirculations" Int. J. for Numerical Methods in Fluids, vol. 12, pp. 203-224, 1991.
15. So, K. L. "Vortex Phenomena in Conical Diffuser" AIAA J., vol5, pp. 1072-1078, 1967.
16. Garg, A. K. and Leibovich, S. "Spectral Characteristics of Vortex Breakdown Flowfields" Phys. Fluids, vol. 22, pp. 2053-2064, 1979.
17. Faler, J. H and Leibovich, S. "Disrupted States of Vortex Flow and Vortex Breakdown" Phys. Fluids, vol. 20, pp. 1385-1400, 1977.
18. Escudier, M. P. and Zehnder, N. "Vortex Flow Regimes" J. Fluid Mech., vol. 115, pp. 105-121, 1982.
19. Stone, H. "Iterative Solution of Implicit Approximations of Multi-Dimensional Partial Differential Equations", SIAM J. Num. Anal., vol. 5, pp 530 - , 1968.

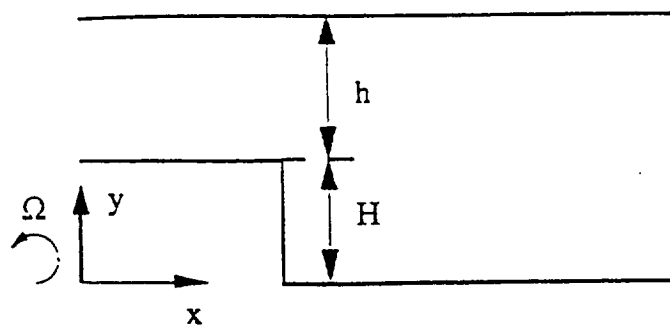
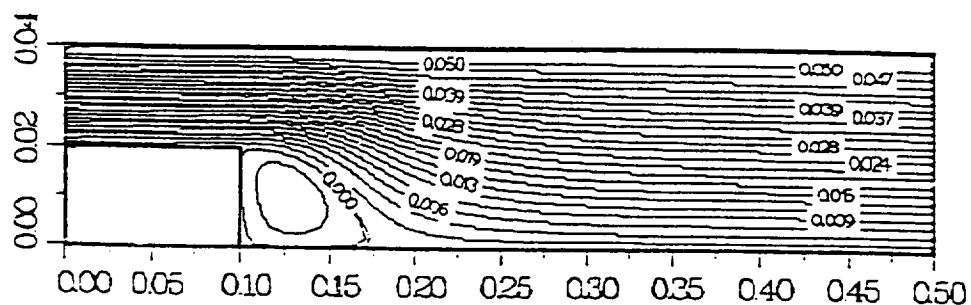
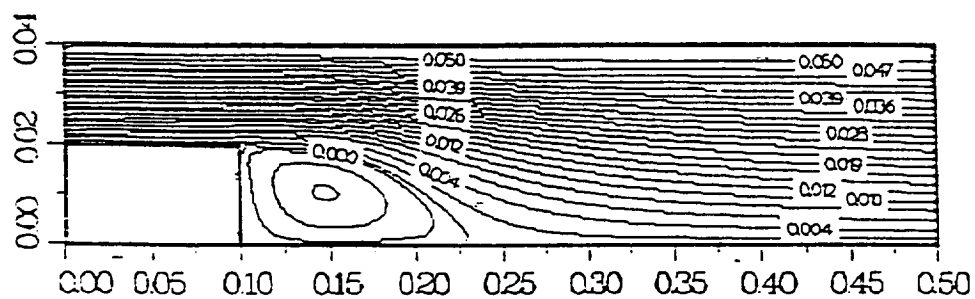


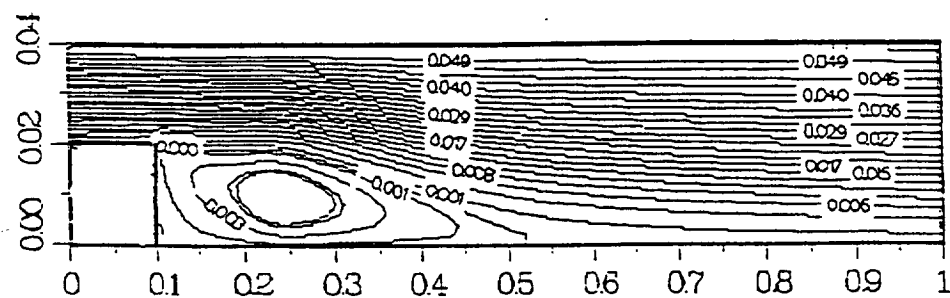
Figure 1. Rotating backward facing step



(a) $Ro=+0.06$

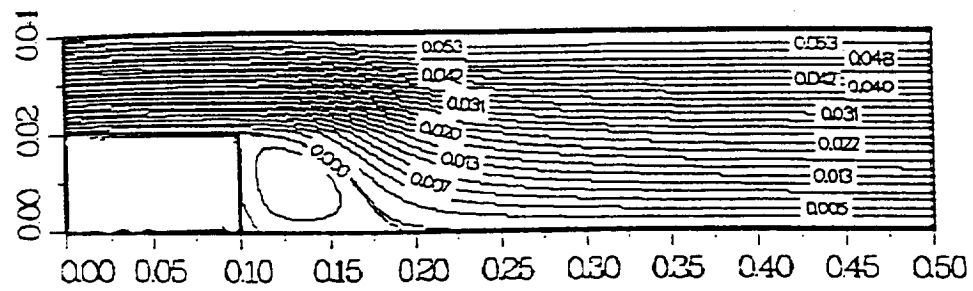


(b) $Ro=0$

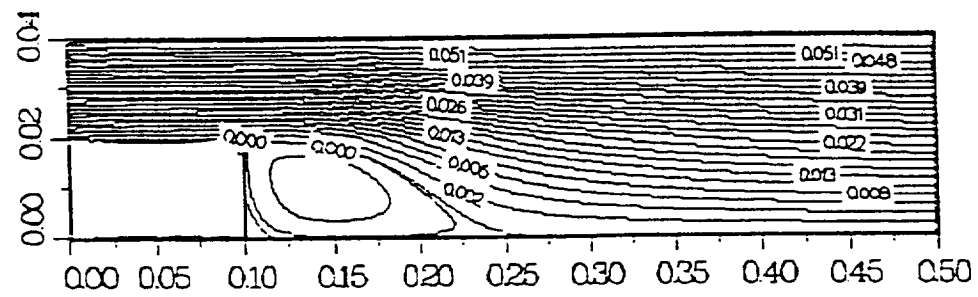


(c) $Ro=-0.06$

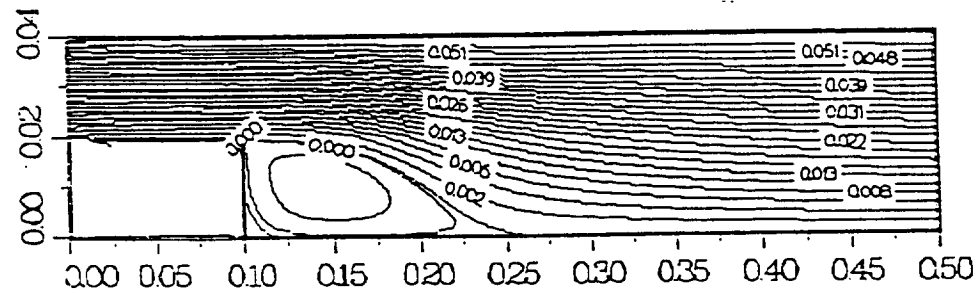
Figure 2. Stream-function contours using wall function near wall treatment



(a) $Ro = +0.06$



(b) $Ro = 0$



(c) $Ro = -0.06$

Figure 3. Stream-function contours using the two-layer near-wall treatment

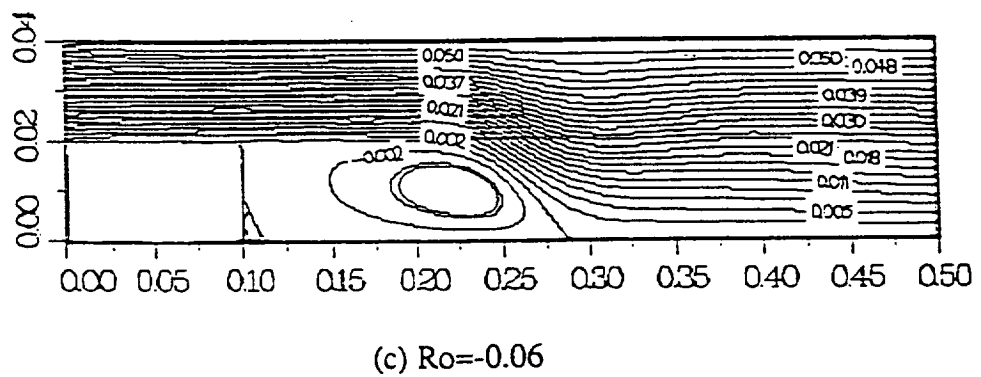
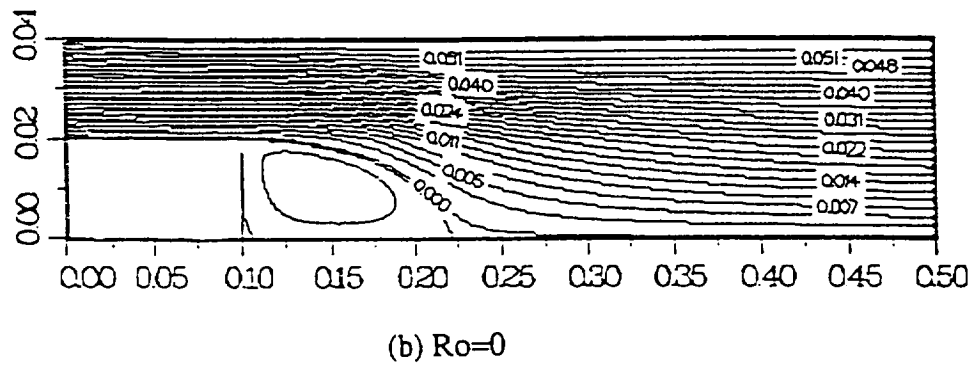
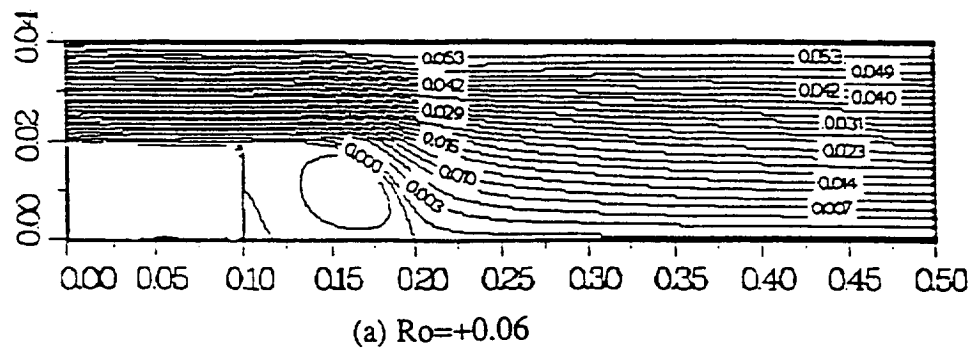


Figure 4. Stream-function contours using the low-Reynolds number near-wall treatment

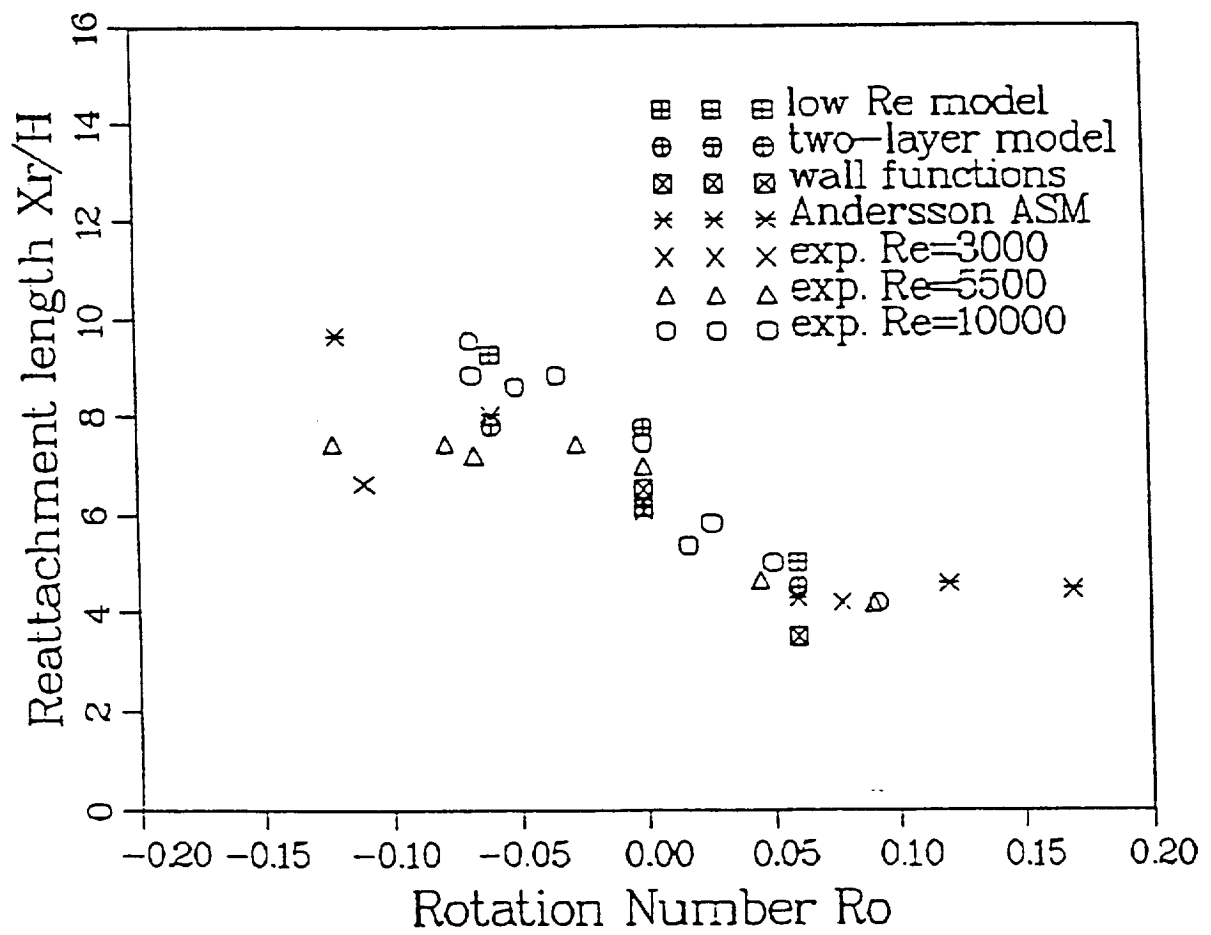


Figure 5. Reattachment length as a function of the rotation number

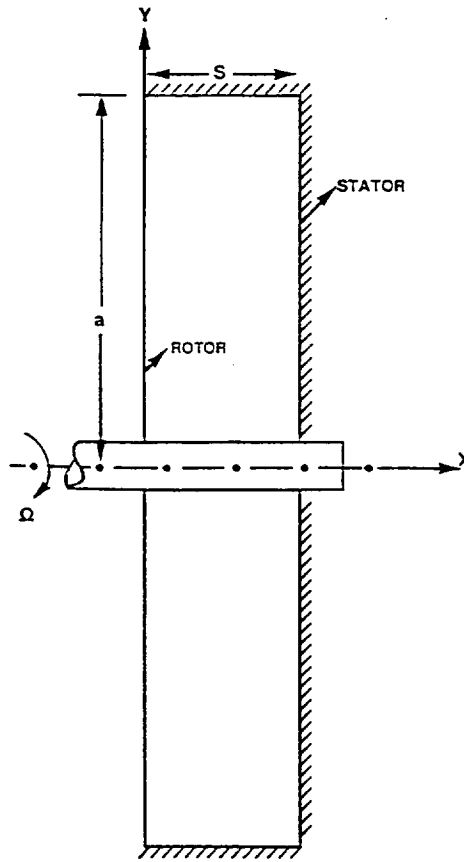


Figure 6. Rotating lid cavity

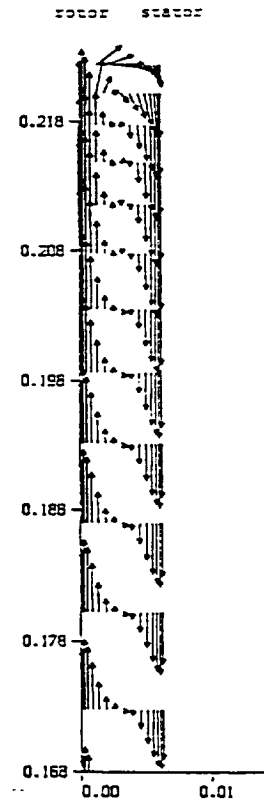


Figure 7. Velocity vectors ($Re = 4.4 \times 10^6$)

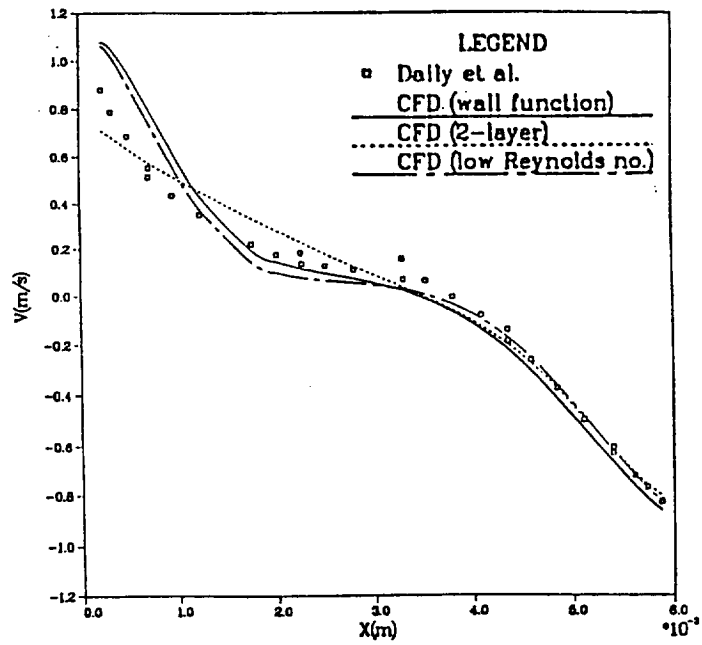


Figure 8. Axial distribution of radial velocity at $r/a = 0.765$

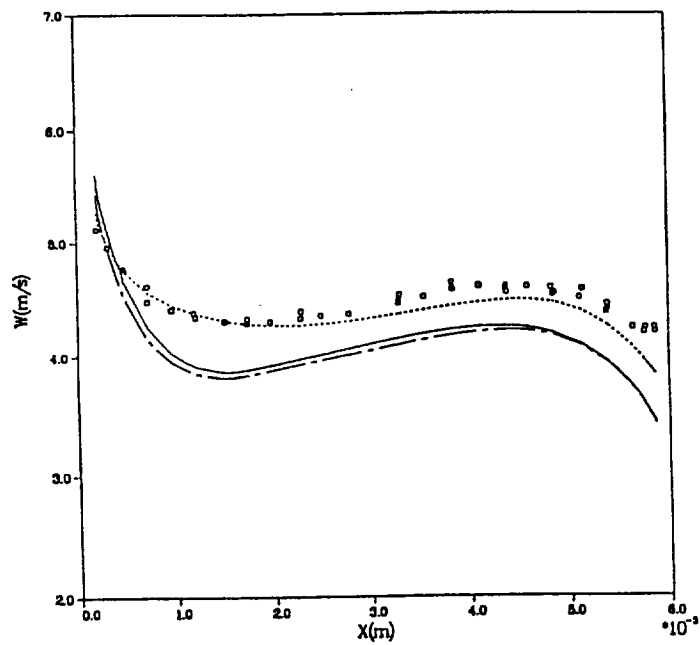


Figure 9. Axial distribution of tangential velocity at $r/a = 0.765$

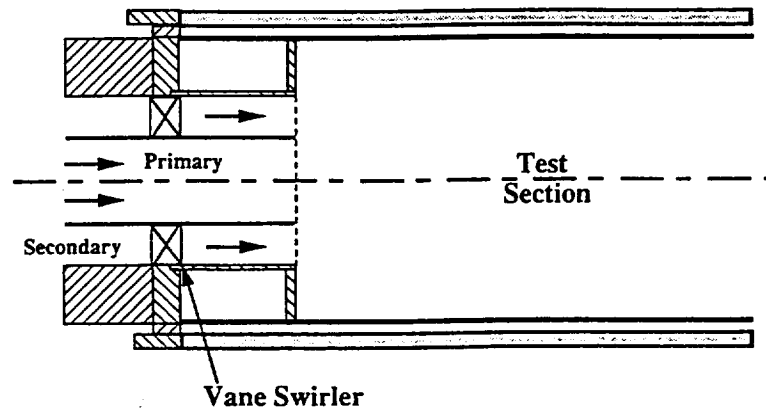


Figure 10. Roback & Johnson's swirling coaxial jets discharging into an expanded duct

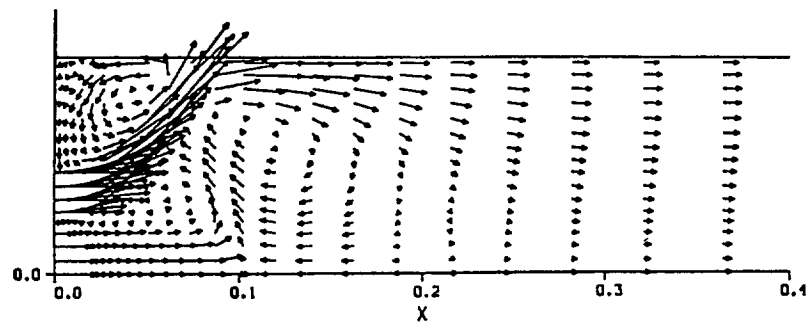


Figure 11. Velocity vectors

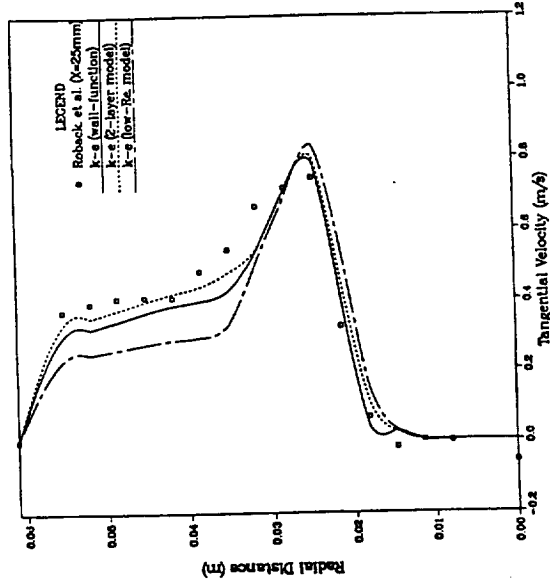


Figure 14. Radial profile of tangential velocity at $x = 0.025m$

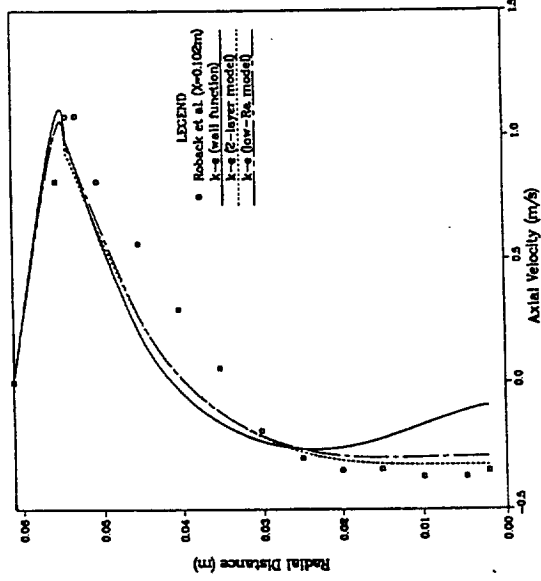


Figure 13. Radial profile of axial velocity at $x = 0.102m$

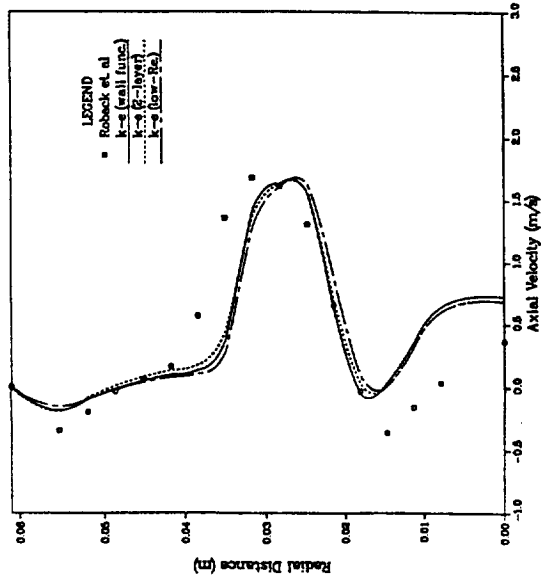


Figure 12. Radial profile of axial velocity at $x = 0.025m$

APPENDIX A

2D/Axisymmetric k - ϵ Turbulence Module Deck

A.1 Introduction

In an attempt to modularize the k - ϵ turbulent physical model -a difficult task as many CFD users may know. A self-contained, stand-alone turbulence module has been constructed that computes turbulent flow quantities using the standard k - ϵ turbulence model. The module is structured to be flexible with options for three near-wall treatments. It can be easily accessed by the user and interfaced with own CFD solvers to calculate turbulent flows.

It is hoped that the program is sufficiently "full proof" and user friendly. However, care must be exercised to identify the limitations of the module to be compatible with the flow solver. Module capabilities and input/output structure is described next in details followed by a FORTRAN listing of the module.

A.2 Program KEMOD

This is basically the solver for the k and ϵ - transport equations. It reads through its argument list different variables from the calling flow solver. These variables are described below where, each variable name ends with either an (I) for Integer variable, (R) for Real variable or (L) for Logical variable.

The flow chart of the program is shown in Figure A.1. It shows the main operations performed by the code.

List of Argument Variable Names

NIMI Number of cell nodes in the I- or ξ -coordinate lines. (input from flow solver)

NJMI Number of cell nodes in the J- or η -coordinate lines. (input from flow solver)

XR	Grid node locations in the x or ξ -direction, dimensioned to XR (NX,NY) (input from flow solver)
YR	Grid node locations in the y or η -direction, dimensioned to YR (NX,NY) (input from flow solver)
UR	Axial or x-direction velocity (u), dimensioned as UR (NX,NY) (input from flow solver)
VR	Radial or y-direction velocity (v), also dimensional as VR (NX,NY) (input from flow solver)
WR	Azimuthal velocity (w), dimensional WR (NX,NY) (input from flow solver)
TER	Turbulence kinetic energy k , dimensioned TER (NX,NY) (calculated in KEMOD and returned to flow solver)
EDR	Turbulent energy dissipation rate ϵ , dimensioned EDR (NX,NY) (calculated in KEMOD and returned to flow solver)
URFKR	Under-relaxation factor for k -equation (input from flow solver)
URFER	Under-relaxation factor for ϵ -equation (input from flow solver)
PRTKR	Prandtl/Schmidt number for turbulent energy-equation, assumed known (input from flow solver)
PRTER	Prandtl/Schmidt number for turbulent energy dissipation equation, assumed known (input from flow solver)
GR	<p>= 1.0 if second order upwinding is desired</p> <p>= 0.0 if first order upwinding is used</p> <p>(input from flow solver. Usually calculation of k and ϵ are not very sensitive to the order of upwinding used)</p>

F1R	Mass flux variable at cell faces in x- or ξ -direction, dimensioned F1R (NX,NY) (input from flow solver)
F2R	Mass flux variable at cell faces in y or η -direction, dimensioned F2R (NX,NY) (input from flow solver)
ITERI	Iteration number (input from flow solver)
VISCOSR	Dynamic viscosity (input from flow solver)
VISR	Eddy viscosity, dimensioned VISR (NX,NY) (calculated in KEMOD and returned to main solver)
AKSIL	Logical variable for axisymmetric geometry (AKSIL= TRUE) or plain geometry (AKSIL= FALSE) (input from flow solver)
LREL	Logical variable for Lam & Bremhorst's low-Reynolds number model (LREL= TRUE) or others (LREL= FALSE) (input from flow solver)
LAY2L	Logical variable for Patel's two-layer model if (LAY2L= TRUE) or others (LAY2L = FALSE) (input from flow solver)
C1R	Turbulence model constant, C_1 (input from flow solver)
C2R	Turbulence model constant, C_2 (input from flow solver)
CMUR	Turbulence model constant, C_μ (input from flow solver)
I2LWI	Grid line location from the west wall in the x-direction for the two-layer model (input from flow solver)
I2LEI	Grid line location from the east wall in the x-direction for the two-layer model (input from flow solver)
J2LSI	Grid line location from the south wall in the y-direction for the two-layer model (input from flow solver)

J2LNI	Grid line location from the north wall in the y-direction for the two-layer model (input from flow solver)
JTBEI	Boundary condition flag along east boundary must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall e.g., for an outlet boundary condition on the east boundary set JTBEI to NJ*2, and similarly for other boundaries, dimensioned JTBEI (NY) (input from flow solver)
JTBWI	Boundary condition flag along west boundary, dimensioned JTBWI (NY) (input from flow solver)
ITBNI	Boundary condition flag along north boundary, dimensioned ITBNI (NX) (input from flow solver)
ITBSI	Boundary condition flag along south boundary, dimensioned ITBSI (NY) (input from flow solver)

Program KEMOD is interfaced with the main flow solver by a call to KEMOD with its arguments. For iterative flow solvers KEMOD is called within the iteration sequence after the solution of the momentum equations where the mean velocities are passed to KEMOD. There are different flow solvers utilizing different schemes from staggered to nonstaggered grid arrangement and for nonorthogonal coordinate system there are at least three alternatives to the choice of the velocity components

- i. Cartesian velocity components
- ii. Contravariant velocity components
- iii. Covariant velocity components

The Cartesian velocity components are the most widely used and have the advantage of simple formulation of the governing equations. Whatever the arrangement used, mass fluxes at cell faces are required and passed to KEMOD as F1R and F2R in both directions. The location of other variables such as k and ϵ are at the cell center or cell nodes.

The module starts by reassigning variable names passed to it from flow solver to names that are shared with the different subroutines of the module in a common statement file "KEMOD·COMMON". Then a check is made if it is the first iteration in which case the grid file "GRIDF" is called -after passing the grid node locations XR & YR in KEMOD- in order to calculate grid related quantities which will be explained later. The need to call GRIDG can be waived if all the grid data are passed to the module. That is all the information about the grid such as interpolation factors FX and FY, cell areas (ARE) and volumes (VOL) and normal distances of first grid point from grid boundaries (DNS from south boundary, DNN - from north boundary, DNW - from west boundary and DNE - from east boundary).

After this a call to subroutine CALCE is made to calculate the turbulent kinetic energy k (with the identifier IPHI=1) followed by a check if the low-Reynolds number model or the two-layer model are to be used in which case subroutine TWOLAY is called. The energy dissipation equation is solved next by a call to subroutine CALCE again with the identifier IPHI=2. The turbulent viscosity is updated next by calling subroutine MODVIS. A brief description of each subroutine is given next.

A.3 Subroutines

GRIDG

Before calling this subroutine, the coordinates of all grid nodes, defined in reference to a fixed Cartesian coordinate frame are read. Figure A.2 shows the position of cell and grid nodes.

This subroutine is called only once to calculate coordinates of grid nodes (intersection of grid lines) and geometrical properties of the grid (cell areas and volumes, interpolation factors, normal distances of near-boundary cell nodes from boundary). All variables including grid node coordinates are converted to one-dimensional arrays. These are formed by scanning the grid in J-direction (figure A.2) for I=1, and then repeating for all I's. The position of any node in one-dimensional array is therefore defined as;

$$IJ = (I,J) = (I-1) * NJ + J$$

The actual number of grid nodes is one row and one column less than for all cell nodes. For I = NI and J = NJ fictitious grid nodes are introduced which have the same coordinates as actual nodes on NI-1 in I-direction and NJ-1 in J-direction.

The subroutine then calculates interpolation factors which are associated with cell nodes and are used in the main program to calculate values of dependent variables at locations other than cell nodes (cell centers). Definition of these are given in Figure A.3. Cell areas and volumes are calculated next followed by calculations of normal distances of near-boundary nodes from all four outer boundaries.

CALCE (PHI, IPHI)

This subroutine solves the linearized and discretized transport equations for the turbulent energy k and the energy dissipation rate ϵ . The two dummy parameters in the calling statement, PHI and IPHI, represent arrays containing dependent variables for which the equation is to be solved. the subroutine sets up the convective and diffusive coefficients over the entire field. Then it calculates the source terms for either k or ϵ transport equations. A call is made to entry MODPHI in order to modify these sources and boundary coefficients to suit the particular problem. Moreover, a check is made if the two-layer model is selected then the energy dissipation is set algebraically in the sublayer region.

The discretized equations have the form

$$A_p \Phi_p = \sum_{i=E,W,N,S} A_i \Phi_i + S\Phi$$

where the coefficients A_i ($i=E,W,N,S$ see figure A.3) contain both the convective and diffusive fluxes. these equations are assembled and solved by calling subroutine SOLSIP which is based on Stone's Strongly Implicit Solver [19].

TWOLAY

This subroutine is called if the two-layer or low-Reynolds number models are used. It calculates the different coefficients needed to describe the energy dissipation and eddy viscosity. In this

subroutine the normal distances used in the definition of the turbulent Reynolds number R_y at corner regions are calculated based on the normal distance nearest to the solid boundary.

SOLSIP

This subroutine solves the system of linear algebraic equations for k and ϵ using Stone's Implicit Procedure [19]. The array RES (IJ) is used to store residuals. The sum of absolute residuals "RESORP" calculated in the first pass through this part of the routine is used as a measure of convergence of the solution process as a whole and this value is stored in RESOR (IPHI). This variable RESOR (IPHI) is passed to the main solver and if desired can be normalized and compared with the maximum error allowed there. If necessary, inner iterations counter L and the sum of absolute residuals RESORP are printed out to monitor the rate of convergence of k and ϵ solution. If the ratio RSM is greater than the maximum allowed for the variable in question, SOR (IPHI), and the number of inner iterations is smaller than a prescribed maximum, NSWP (IPHI), then the routine repeats the sequence of calculating the residuals, increment vectors and updating the dependent variable.

USERM

This subroutine has different ENTRY points or sections where variables are updated and boundary conditions are set.

Section MODVIS

This section calculates effective viscosity (Eq. 8). It is called after calculating k and ϵ . At locations where ϵ is close to zero (i.e., $\leq 10^{-30}$) viscosity is set to zero. A provision is made for under relaxing changes in effective viscosity which may help to stabilize oscillations and improve convergence rate.

Section MODPHI

This section is called from CALCE subroutine and sets the boundary conditions for k and ϵ depending on which variable being called (IDIR = 1 for k and IDIR = 2 for ϵ). For the k -equation, the south boundary is checked first if it is one of four options:

- (1) An inflow boundary ITBS(I) = 1, where the source term is set to accept the inlet values at J = 1 (south boundary)
- (2) Outflow boundary ITBS(I) = 2, where zero gradient in y or η -direction is employed.
- (3) Symmetry boundary, TBS(I) = 3, where gradients normal to symmetry plane are zero.
- (4) Wall boundary, ITBS(I) = 4, where the production term GENTS(I) calculated from subroutine WALLFN in program MODIFY is added to the rest of the source term SU(IJ).

Boundary conditions for the ε -equation are similar to those of k except at the wall where they are set to appropriate values for each near wall treatment.

A.4 Program MODIFY

This program is compiled separately and is called from the u and v solver routines. It basically updates the flux source term of the discretized momentum equation due to wall shear stresses. If the u-momentum equation for example is discretized in the form

$$A_p^* u_p = \sum_{i=EWNS} A_i u_i + S_u^*$$

where P, E, W, N, S are cell nodes as shown in Figure A.3, and A_p^* and A_i 's contain convective and diffusive coefficients. S_u^* is the source term containing pressure gradients and cross-derivative diffusion terms and convective terms for second-order upwinding scheme. This source term is usually linearized as $S_u^* = S_u - B_p u_p$. The term B_p is usually moved to the left hand side of the equation and modifies the diagonal coefficient $A_p = A_p^* + B_p$, and the equation can be written as

$$A_p u_p = \sum_{i=EWNS} A_i u_i + S_u$$

Then S_u and B_p are passed to subroutine MODIFY where they are modified if a wall is present (e.g., ITBS(I) = 4 for south boundary).

For an iterative flow solver using the finite-volume methodology. A typical interface and call to the $k-\epsilon$ module from the main flow solver can be represented by a flow chart as shown in figure A.4.

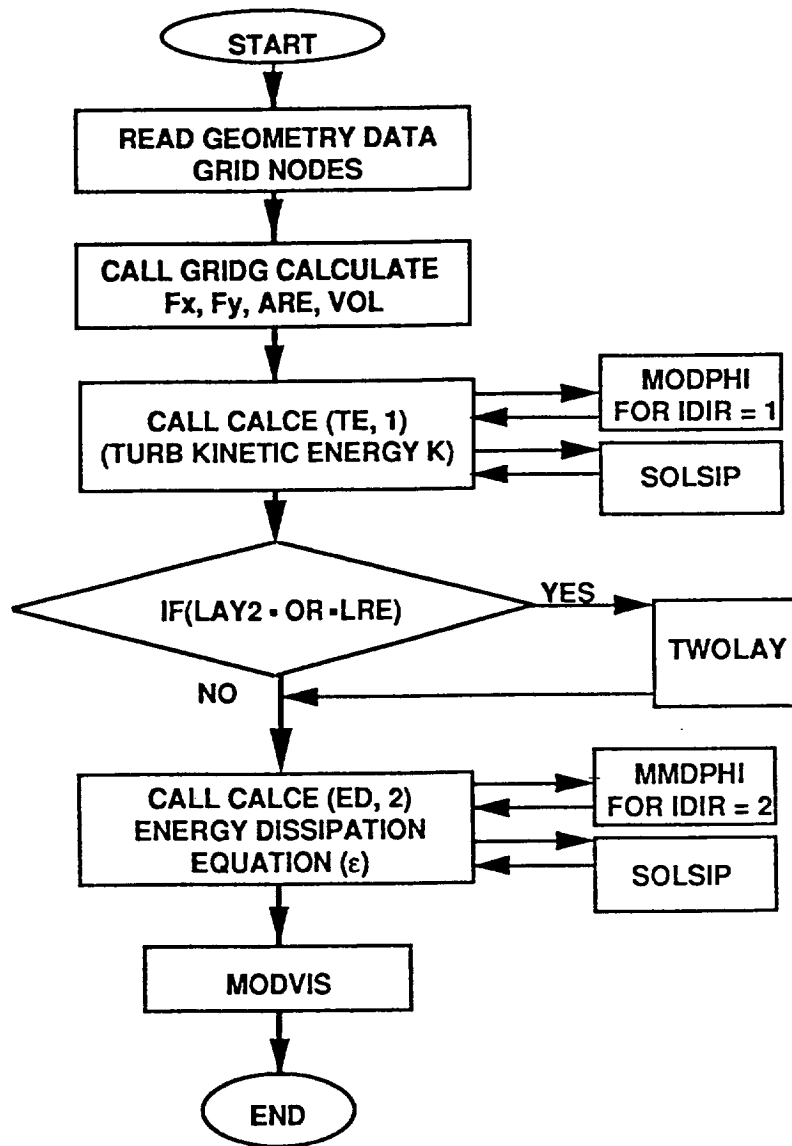


Figure A.1 2D/axisymmetric k - ε module deck flow chart

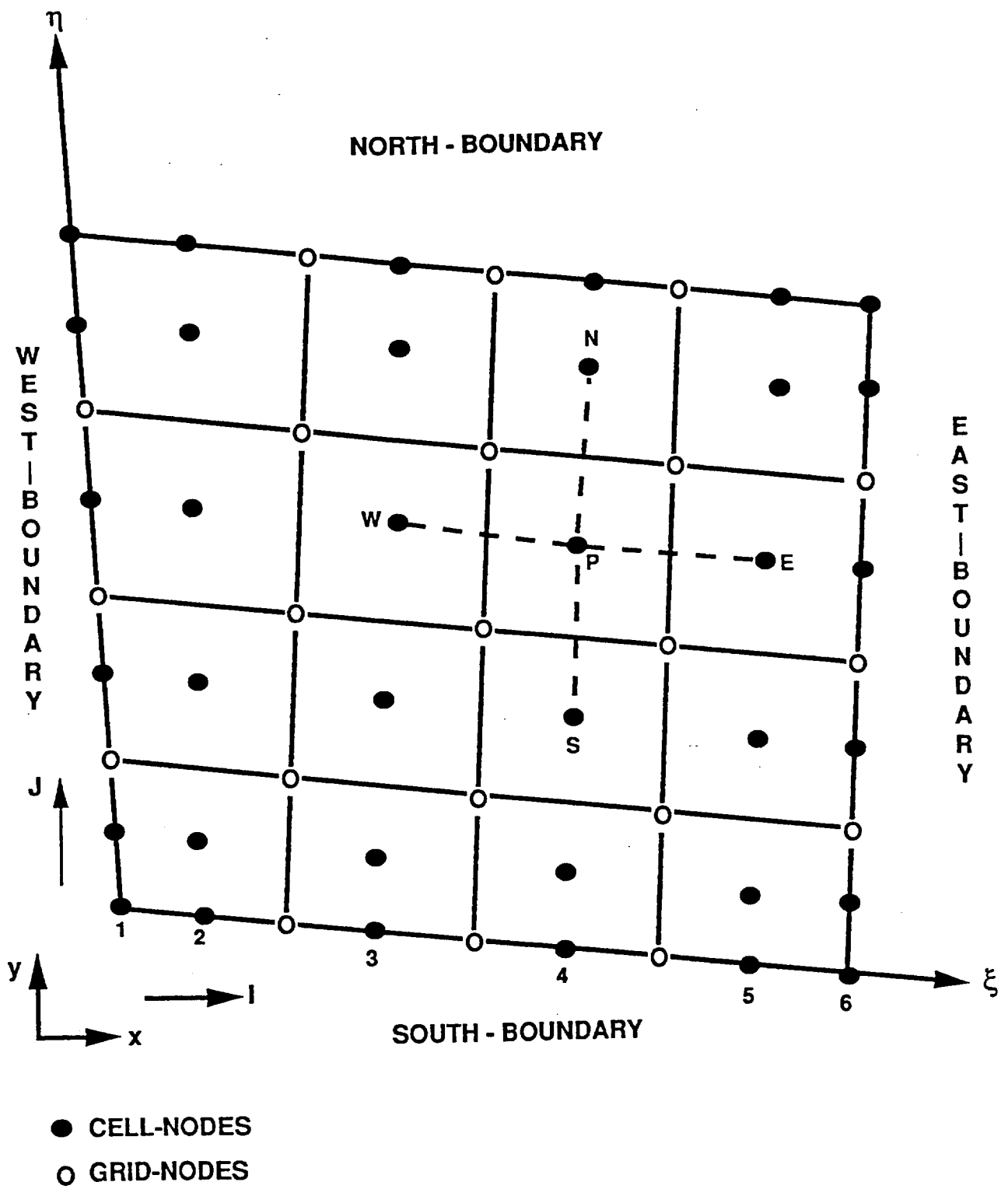
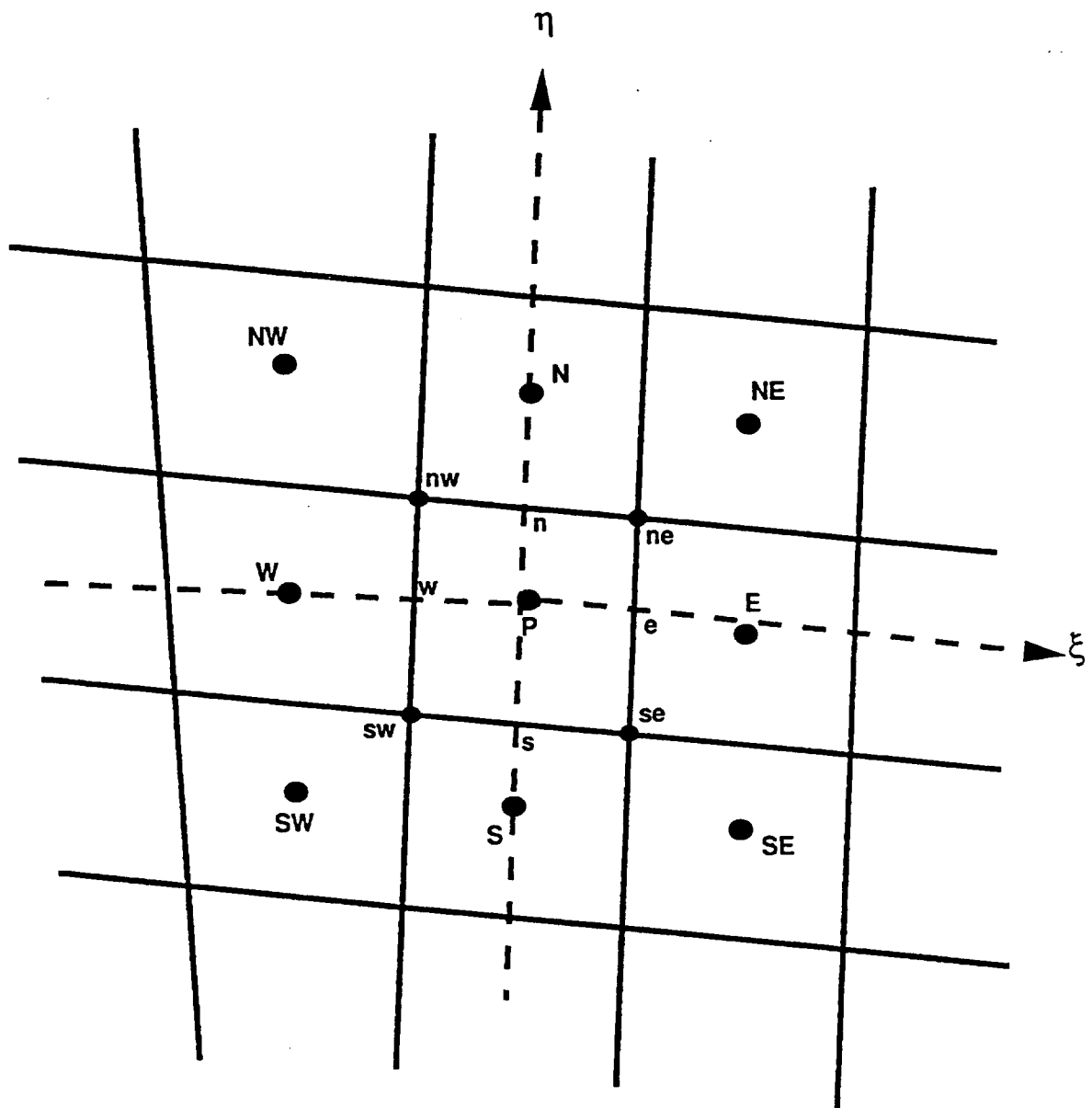


Figure A.2 Position of cell and grid nodes



$$FX_P = \frac{\overline{Pe}}{\overline{Pe} + \overline{eE}}, \quad FY_P = \frac{\overline{Pn}}{\overline{Pn} + \overline{nN}}$$

Figure A.3 Definition of the interpolation factors

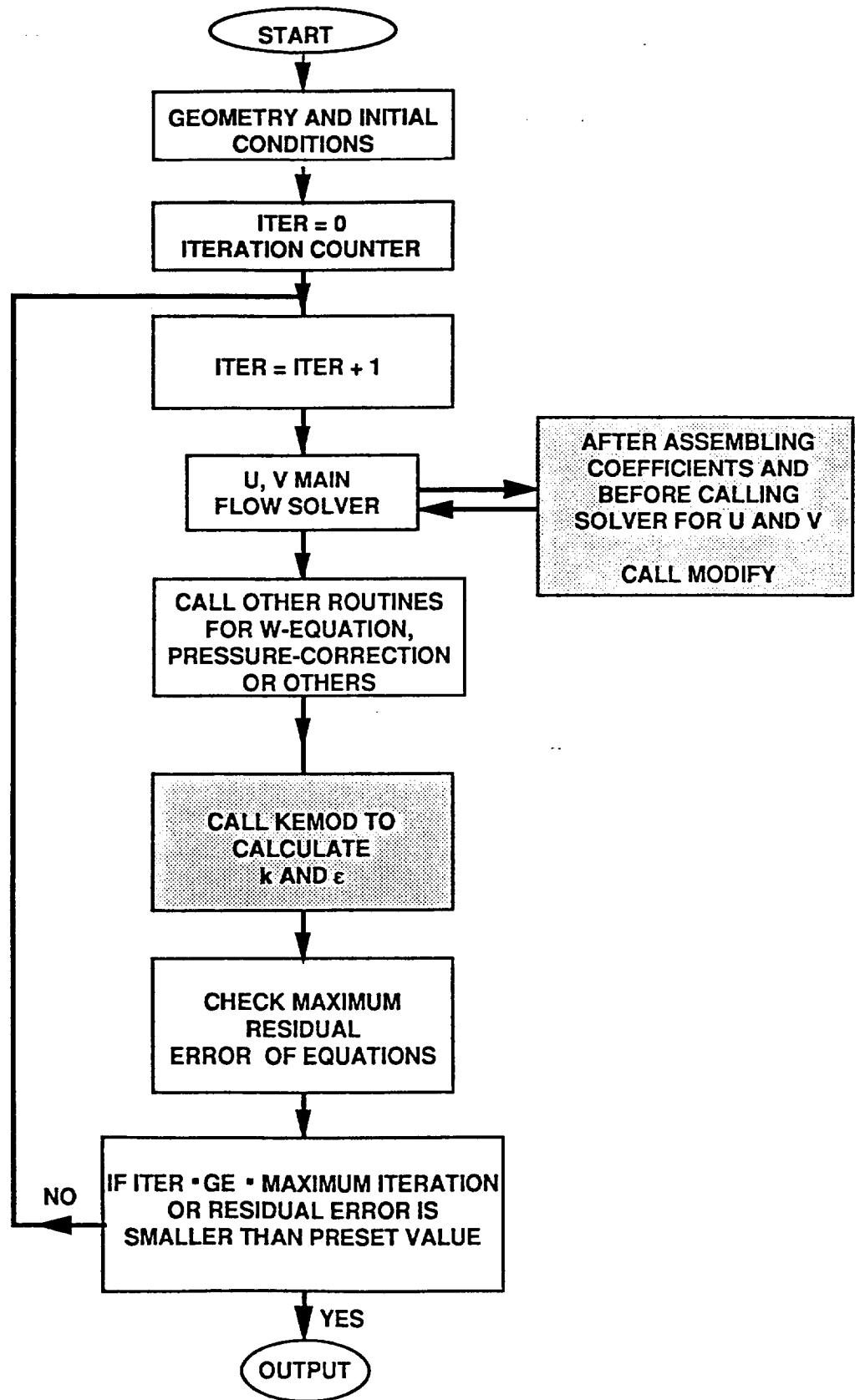


Figure A.4 Typical main flow solver with calls to the 2D/axisymmetric k - ϵ module

```

Oct 12 1996 16:41          skemod_2d          Page 1
1 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
2 C
3 C      2D/AXISYMMETRIC SINGLE-SCALE K-E TURBULENCE MODULE
4 C
5 C      Rocketdyne CFD Technology Center
6 C
7 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
8 C
9 C      Single Scale 2-Equation with 3 Wall Treatments
10 C    1.) Wall Function 2.) Two Layer 3.) Low Reynolds Number
11 C
12 C
13 C-----
14 SUBROUTINE KEMOD (NIMI,NJMI,XR,YR,UR,VX,NX,NY),VR(NX,NY),
15 & URPRK,URPER,PRTKR,PRTER,GR,FIR,F2R,ITERI,VISCOSR,VISR,
16 & AKSIL,LREL,LAY2L,CIR,C2R,CMUR,I2LWI,I2LEI,J2LSI,
17 & J2LNI,JTBEI,JTBWI,ITBNI,ITBSI)
18 C-----
19 INCLUDE 'Kemod.h'
20 C
21 DIMENSION XR(NX,NY),YR(NX,NY),VR(NX,NY),UR(NX,NY),VR(NX,NY),
22 & WR(NX,NY),TER(NX,NY),EDR(NX,NY),FIR(NX,NY),F2R(NX,NY),
23 & VISR(NX,NY),JTBEI(NY),JTBWI(NY),ITBNI(NX),ITBSI(NX)
24 C
25 COMMON/GR/ X(NX,NY),Y(NX,NY)
26 C
27 DATA GREAT,SMALL/1.E30,1.E-30/
28 DATA HAF,QTR/0.5,0.25/
29 DATA SOR/0.1,0.1/
30 DATA NSWP/10,10/
31 DATA WONG/0.0/
32 C
33 C--EQUATE VARIABLES IN ARGUMENT LIST TO THOSE IN COMMON BLOCK
34 C
35 NIM=NIMI
36 NJM=NJMI
37 NI=NIM+1
38 NJ=NJM+1
39 URFK=URFKR
40 URF=URFER
41 PRTE=PRTKR
42 PRTE=PRTER
43 G=GR
44 ITER=ITERI
45 VISCOS=VISCOSR
46 C1=CIR
47 C2=C2R
48 CMU=CMUR
49 C
50 C--LOGICAL VARIABLES
51 C
52 AKSI=AKSIL
53 LRE=LREL
54 LAY2=LAY2L
55 C
56 C--TWO-LAYER TURBULENT PARAMETERS
57 C
58 I2LW=I2LWI
59 I2LE=I2LEI
60 J2LS=J2LSI
61 J2LN=J2LNI
62 C
63 DO 10 I=1,NI
64 IMNJ(I)=(I-1)*NJ
65 CONTINUE
66 C
67 C--BOUNDARY CONDITION IDENTIFIERS
68 C
69 DO 20 I=1,NIM
70 ITBN(I)=ITBNI(I)
71 ITBS(I)=ITBSI(I)

```

Oct 12 1996 16:41 **sskmod_2d** Page 2

```

72 20 CONTINUE
73 C
74 DO 30 J=1,NJM
75 JTBE(J)=JTBEI(J)
76 JTBW(J)=JTBWI(J)
77 30 CONTINUE
78 C
79 C
80 DO 50 I=1,NI
81 DO 50 J=1,NJ
82 IJ=(I-1)*NJ+J
83 X(I,J)=XR(I,J)
84 Y(I,J)=YR(I,J)
85 U(IJ)=UR(I,J)
86 V(IJ)=VR(I,J)
87 W(IJ)=WR(I,J)
88 TE(IJ)=TER(I,J)
89 ED(IJ)=EDR(I,J)
90 VIS(IJ)=VISR(I,J)
91 F1(IJ)=FIR(I,J)
92 F2(IJ)=F2R(I,J)
93 50 CONTINUE
94 C
95 C
96 C----- CALCULATE GRID GEOMETRIC VARIABLES INITIALLY
97 C
98 C IF(ITER.LE.1) THEN
99 CALL GRIDG
100 ENDIF
101
102 C-----
103 C
104 C
105 C--- CALL KINETIC ENERGY SOLVER
106 CALL CALCE ('E,1)
107 C
108 C-----2-LAYER OR LOW-RE. MODELS
109 IF(LAY2.OR.LRE) CALL TWOLAY
110 C
111 C--- CALL ENERGY DISSIPATION SOLVER
112 CALL CALCE (ED,2)
113 C
114 C-----UPDATE AND CALCULATE EDDY VISCOSITY
115 CALL MODVIS
116 C
117 DO 60 I=1,NI
118 DO 60 J=1,NJ
119 IJ=(I-1)*NJ+J
120 TER(I,J) = TE(IJ)
121 EDR(I,J) = ED(IJ)
122 VISR(I,J) = VIS(IJ)
123 60 CONTINUE
124 C
125 RETURN
126 END
127 C
128 C-----
129 C SUBROUTINE GRIDG
130 C
131 C INCLUDE 'kmod.h'
132 C INCLUDE 'kmod.h'
133 C
134 C COMMON/GR/ X(NX,NY), Y(NX,NY)
135 C
136 NJ=NJM+1
137 NI=NIM+1
138 NING=NI*NJ
139 DO 2 I=1,NI
140 INNJ(I)=(I-1)*NJ
141 2 CONTINUE
142 DO 3 I=1,NIM

```

Oct 12 1996 16:41

sskmod_2d

Page 3

```

143 J=NMJ
144 X(I,J+1)=X(I,J)
145 Y(I,J+1)=Y(I,J)
146 3 CONTINUE
147 DO 4 J=1,NJ
148 I=NIM
149 X(I+1,J)=X(I,J)
150 Y(I+1,J)=Y(I,J)
151 4 CONTINUE
152 C
153 C.... GRID ORIGIN AT X=0, Y=0
154 DO 5 I=1,NI
155 DO 5 J=1,NJ
156 IJ=(I-1)*NJ+J
157 XX(IJ)=X(I,J)
158 YY(IJ)=Y(I,J)
159 5 CONTINUE
160 C
161 C-----CALCULATION OF INTERPOLATION FACTORS
162 C
163 DO 6 IJ=1,NINJ
164 FX(IJ)=0.
165 FY(IJ)=0.
166 6 CONTINUE
167 C
168 DO 7 J=2,NJM
169 IJ=J
170 FX(IJ)=0.
171 LIE=NIM-1
172 DO 8 I=2,LIE
173 IJ=IMNJ(I)+J
174 IPJ=IJ+NJ
175 IJM=IJ-1
176 IMJ=IJ-NJ
177 DXF=0.5*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
178 DXP=0.5*(XX(IPJ)-XX(IJ)+XX(IPJ-1)-XX(IJM))
179 DYP=0.5*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
180 DYP=0.5*(YY(IPJ)-YY(IJ)+YY(IPJ-1)-YY(IJM))
181 DXF=SQRT(DXP**2+DYP**2)
182 DFE=SQRT(DXP**2+DYP**2)
183 FX(IJ)=DXF/(DXF+DFE)
184 8 CONTINUE
185 IJ=IMNJ(NIM)+J
186 FX(IJ)=1.0
187 7 CONTINUE
188 C
189 DO 9 I=1,NI
190 IJ=IMNJ(I)+1
191 FX(IJ)=FX(IJ+1)
192 IJ=IMNJ(I)+NJ
193 FX(IJ)=FX(IJ-1)
194 9 CONTINUE
195 C
196 DO 10 I=2,NIM
197 IJ=IMNJ(I)+1
198 FY(IJ)=0.0
199 LJM=NJM-1
200 DO 11 J=2,LJM
201 IJ=IMNJ(I)+J
202 IJP=IJ+1
203 IJM=IJ-1
204 DXF=0.5*(XX(IJ)+XX(IMJ)-XX(IJM)-XX(IMJ-1))
205 DXPN=0.5*(XX(IJP)+XX(IMJ+1)-XX(IJ)-XX(IMJ))
206 DYP=0.5*(YY(IJ)+YY(IMJ)-YY(IJM)-YY(IMJ-1))
207 DYPN=0.5*(YY(IJP)+YY(IMJ+1)-YY(IJ)-YY(IMJ))
208 DEP=SQRT(DXP**2+DYP**2)
209 DEPN=SQRT(DXPN**2+DYPN**2)
210 FY(IJ)=DEP/(DEP+DEPN)
211 11 CONTINUE
212 IJ=IMNJ(I)+NJM
213

```

sskmod_2d

2

Oct 12 1996 16:41

sskmod_2d

Page 4

```

214 FY(IJ)=1.0
215 10 CONTINUE
216 C
217 DO 12 J=1,NJ
218 FY(J)=FY(J+NMJ)
219 IJ=IMNJ(NI)+J
220 FY(IJ)=FY(IJ-NJ)
221 12 CONTINUE
222 C
223 C-----CALCULATION OF CELL AREAS
224 C
225 DO 13 IJ=1,NINJ
226 ARE(IJ)=0.
227 13 CONTINUE
228 C
229 DO 14 I=2,NIM
230 DO 14 J=2,NJM
231 IJ=IMNJ(I)+J
232 DXNWSW=XX(IJ)-XX(IJ-NJ-1)
233 DYNESW=YY(IJ)-YY(IJ-NJ-1)
234 DXNWSE=XX(IJ-NJ)-XX(IJ-1)
235 DYNWSE=YY(IJ-NJ)-YY(IJ-1)
236 ARE(IJ)=0.5*ABS(DXNWSW*DYNWSE+DYNWSE*DYNWSE)
237 14 CONTINUE
238 C
239 C-----NORMAL DISTANCE FROM THE WALL
240 C
241 DO 15 I=1,NI
242 DNS(I)=0.
243 DNN(I)=0.
244 15 CONTINUE
245 C
246 DO 16 J=1,NJ
247 DNM(J)=0.
248 DNE(J)=0.
249 16 CONTINUE
250 C
251 DO 17 I=2,NIM
252 IJ=IMNJ(I)+2
253 IMJ=IJ-NJ
254 DXB=XX(IJ-1)-XX(IMJ-1)
255 DYP=YY(IJ-1)-YY(IMJ-1)
256 DXBP=0.25*(XX(IJ)-XX(IJ-1)+XX(IMJ)-XX(IMJ-1))
257 DYPB=0.25*(YY(IJ)-YY(IJ-1)+YY(IMJ)-YY(IMJ-1))
258 DNS(I)=DELTA(DXB,DYB,DXBP,DYBP)
259 IJ=IMNJ(I)+NJM
260 IMJ=IJ-NJ
261 DXB=XX(IJ)-XX(IMJ)
262 DYP=YY(IJ)-YY(IMJ)
263 DXBP=0.25*(XX(IJ-1)-XX(IJ-1)+XX(IMJ-1)-XX(IMJ))
264 DYPB=0.25*(YY(IJ-1)-YY(IJ-1)+YY(IMJ-1)-YY(IMJ))
265 DNN(I)=DELTA(DXB,DYB,DXBP,DYBP)
266 17 CONTINUE
267 C
268 DO 18 J=2,NJM
269 IJ=IMNJ(2)+J
270 IMJ=IJ-NJ
271 DXB=XX(IMJ)-XX(IMJ-1)
272 DYP=YY(IMJ)-YY(IMJ-1)
273 DXBP=0.25*(XX(IJ)-XX(IMJ)+XX(IJ-1)-XX(IMJ-1))
274 DYPB=0.25*(YY(IJ)-YY(IMJ)+YY(IJ-1)-YY(IMJ-1))
275 DNN(J)=DELTA(DXB,DYB,DXBP,DYBP)
276 IJ=IMNJ(NIM)+J
277 IMJ=IJ-NJ
278 DXB=XX(IJ)-XX(IJ-1)
279 DYP=YY(IJ)-YY(IJ-1)
280 DXBP=0.25*(XX(IMJ)-XX(IJ)+XX(IMJ-1)-XX(IJ-1))
281 DYPB=0.25*(YY(IMJ)-YY(IJ)+YY(IMJ-1)-YY(IJ-1))
282 DNE(J)=DELTA(DXB,DYB,DXBP,DYBP)
283 18 CONTINUE
284 C

```

Oct 12 1996 16:41

sskmod_2d

Page 5

```

285 C----- CALCULATE CELL VOLUMES
286 C
287 DO 19 IJ=1,NINJ
288 VOL(IJ)=ARE(IJ)
289 19 CONTINUE
290 C
291 IF(AKSI) THEN
292 SIXR=1./6.
293 DO 20 I=2,NIM
294 IY=IMNJ(I)
295 DO 21 J=2,NJM
296 IJ=I+J
297 IMJ=IJ-NJ
298 IMJM=IMJ-1
299 IJM=IJ-1
300 RIJ=YY(IJ)**2
301 RIMJ=YY(IMJ)**2
302 RIMM=YY(IMM)**2
303
304 VOL(IJ)=SIXR*((XX(IJ)-XX(IMJ))* (RIJ+RIMJ+YY(IJ)*YY(IMJ)) +
& (XX(IMJ)-XX(IMM))* (RIMJ+RIMM+YY(IMJ)*YY(IMM)) +
& (XX(IMM)-XX(IJM))* (RIMM+RIJ+YY(IJM)*YY(IJM)) +
& (XX(IJM)-XX(IJ))* (RIJM+RIJ+YY(IJM)*YY(IJ)))
307 &
308 21 CONTINUE
309
310 20 CONTINUE
311 C
312 C----- INITIALIZE VARIABLES INITIALLY
313 C
314 HAF=0.5
315 OTR=0.25
316 SMALL=1.E-30
317 GREAT=1.E30
318 DO 22 IJ=1,NINJ
319 DEN(IJ)=DENSIT
320 VIS(IJ)=VISCOS
321 FMU(IJ)=1.0
322 FLR1(IJ)=1.0
323 FLR2(IJ)=1.0
324 APV(IJ)=0.0
325 AE(IJ)=0.0
326 AS(IJ)=0.0
327 AN(IJ)=0.0
328 AW(IJ)=0.0
329 BE(IJ)=0.0
330 BW(IJ)=0.0
331 BN(IJ)=0.0
332 BS(IJ)=0.0
333 RES(IJ)=0.0
334 R(IJ)=1.0
335
336 22 CONTINUE
337 IF(AKSI) THEN
338 DO 23 IJ=1,NINJ
339 R(IJ)=YY(IJ)
340 23 CONTINUE
341 ENDIF
342 C
343 RETURN
344 END
345 C
346 C----- SUBROUTINE CALCE (PHI,IPHI)
347
348 C-----
349 INCLUDE 'kmod.h'
350 C
351 DIMENSION UW(NY), VW(NY), WW(NY), PHI(NXNY), FXWW(NY), DW(NY)
352 C
353 IF(IPHI.EQ.1) THEN
354 URPFI=1./URFE
355 PRTINVP=1./PRTE

```

sskmod_2d

3

Oct 12 1996 16:41

sskmod_2d

Page 6

```

356 ELSE
357 URPFI=1./URFE
358 PRTINVP=1./PRTE
359 ENDIF
360 C
361 IJ=1
362 PHINE=PHI(IJ)
363 PHINW(IJ)=PHINE
364 C
365 DO 30 J=2,NJM
366 IJ=J
367 IJM=IJ-1
368 IJ=IJ+NJ
369 IJP=IJ+1
370 FYN=FY(IJ)
371 FYS=1.0-FYN
372 AREE=HAF*(ARE(IJ)+ARE(IPJ))
373 DXE=XX(IJ)-XX(IJM)
374 DYE=YY(IJ)-YY(IJM)
375 VIST=VIS(IJ)-VISCOS
376 GAME=HAF*(VISCOS+VIST*PRTINVP)*(R(IJ)+R(IJM))
377 DW(IJ)=GAME/AREE*(DXE**2+DYE**2)
378 PHISE=PHINE
379 PHINE=PHI(IJ+1)*FYN+PHI(IJ)*FYS
380 PHINW(J)=PHINE
381 UW(J)=U(IJ)
382 VW(J)=V(IJ)
383 WW(J)=W(IJ)
384 SNSW(J)=0.
385 FXWW(J)=1.0
386 IF(JTBW(J).EQ.3.OR.JTBW(J).EQ.4) GO TO 30
387 DXKS=QTR*(XX(IPJ)+XX(IPJ-1)-XX(IJ)-XX(IJM))
388 DYKS=QTR*(YY(IPJ)+YY(IPJ-1)-YY(IJ)-YY(IJM))
389 SNSW(J)=-GAME/AREE*(DXKS*DXE+DYKS*DYE)*(PHINE-PHISE)
390 CONTINUE
391 C
392 DO 32 I=2,NIM
393 J=1
394 IJ=IMNJ(I)+J
395 IMJ=IJ-NJ
396 IJP=IJ+1
397 FXP=FX(IJ)
398 FXW=1.-FX(IJ)
399 AREN=HAF*(ARE(IJ)+ARE(IJP))
400 DXN=XX(IJ)-XX(IMJ)
401 DYN=YY(IJ)-YY(IMJ)
402 VIST=VIS(IJ)-VISCOS
403 GAMN=HAF*(VISCOS+VIST*PRTINVP)*(R(IJ)+R(IMJ))
404 DN=GAMN/AREN*(DXN**2+DYN**2)
405 FVSS=1.0
406 PHINE=PHI(IJ+NJ)*FXE+PHI(IJ)*FXW
407 UN=U(IJ)
408 VN=V(IJ)
409 WN=W(IJ)
410 SEWN=0.
411 IF(ITBS(I).EQ.3.OR.ITBS(I).EQ.4) GO TO 33
412 DXET=QTR*(XX(IJP)+XX(IJP-NJ)-XX(IJ)-XX(IMJ))
413 DYET=QTR*(YY(IJP)+YY(IJP-NJ)-YY(IJ)-YY(IMJ))
414 SEWN=-GAMN/AREN*(DXN*DXET+DYN*DYET)*(PHINE-PHINW(J))
415 SEWNBS=SEWN
416 CONTINUE
417 33
418 PHINW(J)=PHINE
419 C
419 C-----THE MAIN LOOP - ASSEMBLY OF COEFFICIENTS AND SOURCES
420 C
421 DO 34 J=2,NJM
422 IJ=IMNJ(I)+J
423 IJP=IJ+NJ
424 IMJ=IJ-NJ
425 IJP=IJ+1
426 IJM=IJ-1

```

Oct 12 1996 16:41 **sskmod_2d** Page 7

```

427 FXE=FX(IJ)
428 FXW=1.-FXE
429 FYN=FY(IJ)
430 FYS=1.-FYN
431 DXE=XX(IJ)-XX(IJM)
432 DYE=YY(IJ)-YY(IJM)
433 DXN=XX(IJ)-XX(IMJ)
434 DYN=YY(IJ)-YY(IMJ)
435 AREE=HAF*(ARE(IJ)+ARE(IPJ))
436 AREN=HAF*(ARE(IJ)+ARE(IPJ))
437 VISE=VIS(IJ)*FXW+VIS(IPJ)*FXE
438 VISE=VISE-VISCOS
439 GAME=HAF*(VISCOS+VISE*PRTINVP)*(R(IJ)+R(IJM))
440 VISN=VIS(IJ)*FYS+VIS(IPJ)*FYN
441 VISNT=VISN-VISCOS
442 GAWN=HAF*(VISCOS+VISNT*PRTINVP)*(R(IJ)+R(IMJ))
443 C
444 DS=DN
445 DE=GAME/AREE*(DXE**2+DYE**2)
446 DN=GAWN/AREN*(DXN**2+DYN**2)
447 C
448 C LINEAR UPWIND DIFFERENCING
449 C
450 AEE=MIN(F1(IJ),0.0)*FX(IPJ)*G
451 AEW=-MAX(F1(IMJ),0.0)*(1.0-FXW(J))*G
452 AEL=-MIN(F1(IMJ),0.0)*FXE*G
453 AEW=-MAX(F1(IJ),0.0)*(1.0-FX(IMJ))*G
454 ANN=MIN(F2(IJ),0.0)*FY(IPJ)*G
455 ASS=-MAX(F2(IJM),0.0)*(1.0-FYSS)*G
456 ANI=-MIN(F2(IJM),0.0)*FYN*G
457 ASI=MAX(F2(IJ),0.0)*(1.0-FY(IJM))*G
458 C
459 AW(IJ)=DW(J)+MAX(F1(IMJ),0.0)-AWW
460 AE(IJ)=DE-MIN(F1(IJ),0.0)-AEE
461 AS(IJ)=DS+MAX(F2(IJM),0.0)-ASS
462 AN(IJ)=DN-MIN(F2(IJ),0.0)-ANN
463 C
464 DXKS=QTR*(XX(IPJ)-XX(IMJ)+XX(IPJ-1)-XX(IMJ-1))
465 DYKS=QTR*(YY(IPJ)-YY(IMJ)+YY(IPJ-1)-YY(IMJ-1))
466 DXET=QTR*(XX(IPJ)-XX(IJM)+XX(IPJ-NJ)-XX(IJM-NJ))
467 DYET=QTR*(YY(IPJ)-YY(IJM)+YY(IPJ-NJ)-YY(IJM-NJ)+2.0*STAG)
468 C
469 PHISE=PHINE
470 PHINE=(PHI(IJ)*FYS+PHI(IPJ)*FYN)*FXW+
471 (PHI(IPJ)*FYS+PHI(IPJ+1)*FYN)*FXE
472 SEWN=SEWN
473 SEWN=-GAME/AREN*(DXN*DXET+DYN*DYET)*(PHINE-PHINW(J))
474 SNSE=-GAME/AREE*(DXKS*DXE+DYKS*DYE)*(PHINE-PHISE)
475 IF(I.EQ.NIM.AND.(JTBE(J).EQ.3.OR.JTBE(J).EQ.4)) SNSE=0.
476 IF(J.EQ.NJM.AND.(ITBN(I).EQ.3.OR.ITBN(I).EQ.4)) SEWN=0.
477 APV(IJ)=SNSE-SNSW(J)+SEWN-SEWS
478 C
479 C LINEAR UPWIND DIFFERENCING
480 C
481 IMJ1=IMJ-NJ
482 IMJ2=MAX(1,IMJ1)
483 APV(IJ)=APV(IJ)+ABE*PHI(IPJ+NJ)+AMW*PHI(IMJ2)+ANN
484 (PHI(IPJ+1)+ASS*PHI(IJM-1))
485 (ABE1*PHI(IPJ)+AW1*PHI(IMJ)+AN1*PHI(IPJ)+ASI*PHI(IJM)
486 APV(IJ)=ABE+AMW+ANN+ASS+AE1+AW1+AN1+ASI
487 C
488 VISS=VIS(IJ)-VISCOS+SMALL
489 C
490 GO TO (37,38) IPHI
491 C
492 C-----TURBULENT KINETIC ENERGY SOURCE TERMS
493 C
494 C 37 CONTINUE
495 DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
496 DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
497 DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))

```

sskmod_2d

Oct 12 1996 16:41 **sskmod_2d** Page 8

```

498 DYSN=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
499 RP=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IMJ-1))
500 US=UN
501 VS=VN
502 WS=WN
503 UN=U(IPJ)*FYN+U(IJ)*FYS
504 VN=V(IPJ)*FYN+V(IJ)*FYS
505 WN=W(IPJ)*FYN+W(IJ)*FYS
506 UP=U(IPJ)*FXE+U(IJ)*FXW
507 VE=V(IPJ)*FXE+V(IJ)*FXW
508 WE=W(IPJ)*FXE+W(IJ)*FXW
509 DUEN=UE-UW(J)
510 DUNS=UN-US
511 DVEN=VE-VW(J)
512 DVNS=VN-VS
513 DWEN=WE-WN(J)
514 DWNS=WN-WS
515 GTERM=(2.0*((DUEN*DYNS-DUNS*DYEW)**2+
516 (DYSN*DXEW-DVNS*DXNS)**2)+(DUNS*DXEW-DUEN*DXNS+
517 DVEN*DYNS-DVNS*DYEW)**2)/(ARE(IJ)**2)
518 IF (AKSI)
519 & GTERM=GTERM+((DYNS*DWEN-DYEW*DWNS)**2+
520 (DXEN*DWNS-DXNS*DWEN-(W(IJ)/RP)*ARE(IJ))**2)
521 & /((ARE(IJ)**2)
522 & +2.0*(V(IJ)/RP)**2)
523 C
524 DIVUV=(DUEN*DYNS-DUNS*DYEW+DVNS*DXEW-DVEN*DXNS)/ARE(IJ)
525 GTERM=GTERM-2./3.*DIVUV**2
526 GTERM=MAX(GTERM,0.)
527 C
528 C GENERATION OF KINETIC ENERGY
529 C
530 GEN(IJ)=(VIS(IJ)-VISCOS)*GTERM
531 SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
532 C
533 BP(IJ)=APV(IJ)+VOL(IJ)*CMU*DEN(IJ)*TE(IJ)*FMU(IJ)/
534 VISS
535 C
536 IF(WONG.NE.0.) THEN
537 DUDY(IJ)=(DUNS*DXEW-DUEN*DXNS)/ARE(IJ)
538 DVDX(IJ)=(DVEN*DYNS-DVNS*DYEW)/ARE(IJ)
539 ENDIF
540 C
541 GO TO 40
542 C
543 C-----DISSIP. OF TURB. KIN. ENERGY SOURCE TERMS
544 C
545 C 38 CONTINUE
546 C
547 SU(IJ)=APV(IJ)+FLR1(IJ)*C1*CMU*GEN(IJ)*DEN(IJ)*TE(IJ)*VOL(IJ)
548 & *FMU(IJ)/VISS
549 BP(IJ)=APV(IJ)+FLR2(IJ)*FMU(IJ)*C2*CMU*DEN(IJ)**2*TE(IJ)*VOL(IJ)
550 & /VISS
551 C
552 RI=0.
553 IF(WONG.NE.0.) THEN
554 VOR=DUDY(IJ)-DVDX(IJ)
555 RI=(0.4*VOR-0.8*WONG)*WONG*(TE(IJ)/(ED(IJ)+SMALL))**2
556 END IF
557 FLR2R=FLR2(IJ)+RI
558 BP(IJ)=APV(IJ)+(FLR2R*C2*CMU*FMU(IJ)*DEN(IJ)*DEN(IJ)
559 & *TE(IJ)/VISS
560 C
561 C 40 CONTINUE
562 UW(J)=UE
563 VW(J)=VE
564 WW(J)=WE
565 SNSW(J)=SNSE
566 PHINW(J)=PHINE
567 FYS=FY(IJM)
568 FXW(J)=FX(IMJ)

```


Oct 12 1996 16:41 **sskmod_2d** Page 9

```

569 C      DW(J)=DE
570 C
571 34 CONTINUE
572 32 CONTINUE
573 C
574 C-----PROBLEM MODIFICATIONS - BOUNDARY CONDITIONS
575 C
576 C      IDIR=IPHI
577 C      CALL MODPHI
578 C
579 C      IF (IPHI.EQ.2 .AND. LAY2 ) THEN
580 C          DO 41 I=2,NIM
581 C              IF (ITBS(I).NE.4) GO TO 42
582 C          DO 43 J=2,J2LS
583 C              IJ=IMNJ(I)+J
584 C              SU(IJ)=GREAT*ED(IJ)
585 C              BP(IJ)=GREAT
586 C          CONTINUE
587 C          IF (ITBN(I).NE.4) GO TO 41
588 C          DO 44 J=J2LN,NJM
589 C              IJ=IMNJ(I)+J
590 C              SU(IJ)=GREAT*ED(IJ)
591 C              BP(IJ)=GREAT
592 C          CONTINUE
593 C          CONTINUE
594 C
595 C      DO 50 J=2,NJM
596 C          IF (JTBW(J).NE.4) GO TO 51
597 C          DO 52 I=2,I2LW
598 C              IJ=IMNJ(I)+J
599 C              SU(IJ)=GREAT*ED(IJ)
600 C              BP(IJ)=GREAT
601 C          CONTINUE
602 C          IF (JTBW(J).NE.4) GO TO 50
603 C          DO 53 I=I2LE,NIM
604 C              IJ=IMNJ(I)+J
605 C              SU(IJ)=GREAT*ED(IJ)
606 C              BP(IJ)=GREAT
607 C          CONTINUE
608 C          CONTINUE
609 C          ENDIF
610 C
611 C      DO 60 I=2,NIM
612 C          DO 60 J=2,NJM
613 C              IJ=IMNJ(I)+J
614 C              AP(IJ)=AM(IJ)+AE(IJ)+AN(IJ)+AS(IJ)+BP(IJ)
615 C              AP(IJ)=AP(IJ)+URFPHI
616 C              SU(IJ)=SU(IJ)+(1.-URF(IPHI))*AP(IJ)*PHI(IJ)
617 C          CONTINUE
618 C
619 C-----SOLVING F.D. EQUATIONS
620 C
621 C      CALL SOLSIP(PHI,IPHI)
622 C
623 C      RETURN
624 C      END
625 C
626 C-----SUBROUTINE TWOLAY
627 C
628 C-----INCLUDE 'kmod.h'
629 C
630 C
631 C
632 C-----ALONG THE SOUTH BOUNDARY
633 C
634 C      DO 70 I=2,NIM
635 C          IF (ITBS(I).NE.4) GO TO 70
636 C          DISNS=GREAT
637 C          DISNW=GREAT
638 C          DISNE=GREAT
639 C

```

Oct 12 1996 16:41 **sskmod_2d** Page 10

```

640 DO 71 J=2,J2LS
641 C      IJ=IMNJ(I)+J
642 C      IJW=IMNJ(I)+1
643 C      IMJW=IJW-NJ
644 C      DXB=XX(IJW)-XX(IMJW)
645 C      DYB=YY(IJW)-YY(IMJW)
646 C      XPW=HAF*(XX(IJW)+XX(IMJW))
647 C      YPW=HAF*(YY(IJW)+YY(IMJW))
648 C      XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
649 C      YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
650 C      DXBP=XBP-XPW
651 C      DYBP=YBP-YPW
652 C      DISNS=DELTA(DXB,DYB,DXBP,DYBP)
653 C-----CHECK WEST BOUNDARY
654 C      IF (JTBW(J).EQ.4) THEN
655 C          IJW=J
656 C          IMJW=IJW-1
657 C          DXB=XX(IJW)-XX(IMJW)
658 C          DYB=YY(IJW)-YY(IMJW)
659 C          XB=HAF*(XX(IJW)+XX(IMJW))
660 C          YB=HAF*(YY(IJW)+YY(IMJW))
661 C          XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
662 C          YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
663 C          DXBP=XBP-XB
664 C          DYBP=YBP-YB
665 C          DISNW=DELTA(DXB,DYB,DXBP,DYBP)
666 C          ENDIF
667 C-----CHECK EAST BOUNDARY
668 C      IF (JTBW(J).EQ.4) THEN
669 C          IJW=IMNJ(NIM)+J
670 C          IMJW=IJW-1
671 C          DXB=XX(IJW)-XX(IMJW)
672 C          DYB=YY(IJW)-YY(IMJW)
673 C          XB=HAF*(XX(IJW)+XX(IMJW))
674 C          YB=HAF*(YY(IJW)+YY(IMJW))
675 C          XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
676 C          YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
677 C          DXBP=XBP-XB
678 C          DYBP=YBP-YB
679 C          DISNE=DELTA(DXB,DYB,DXBP,DYBP)
680 C          ENDIF
681 C
682 C      DISN=MIN(DISNS,DISNW,DISNE)
683 C      RK=DISN*DEN(IJ)*SQRT(TE(IJ))/VISCOS
684 C      IF (LAY2) THEN
685 C          ALMU=C11*DISN*(1.0-EXP(-RK/ALMU))
686 C          ALD=C11*DISN*(1.0-EXP(-RK/AED))
687 C          IF (WOMG.NE.0.) THEN
688 C              AME=1.0+1.30*(0.40*DUDY(IJ)-0.80*WOMG)*WOMG*
689 C              # (TE(IJ)/(ED(IJ)*SMALL))**2
690 C          AME=ABS(AME)
691 C          ALMU=ALMU*AME**1.5
692 C          ALD=ALD*AME**0.5
693 C          END IF
694 C          ED(IJ)=SORT(TE(IJ))**3/ALD
695 C          VIS2(IJ)=VISCOS*DEN(IJ)*CMU*SQRT(TE(IJ))*ALMU
696 C          ELSE
697 C              RT=DEN(IJ)*TE(IJ)*TE(IJ)/(VISCOS*ED(IJ))
698 C              FMU(IJ)=(1.0+20.5/RT)*(1.0-EXP(-0.0165*RT))**2
699 C              FLR1(IJ)=1.0*(0.05/FMU(IJ))**3
700 C              FLR2(IJ)=1.0-EXP(-RT*RT)
701 C          ENDIF
702 C          CONTINUE
703 C          CONTINUE
704 C
705 C-----ALONG THE NORTH BOUNDARY
706 C
707 C      DO 72 I=2,NIM
708 C          IF (ITBN(I).NE.4) GO TO 72
709 C          DISNW=GREAT
710 C          DISNE=GREAT

```

Oct 12 1996 16:41	sskmod_2d	Page 11
711	DISN= GREAT	
712	DO 73 J=J2LN, NJM	
713	IJ=IMNJ(I)+J	
714	IJW=IMNJ(I)+NJM	
715	IMJW=IJW-NJ	
716	DXB=XX(IJW)-XX(IMJW)	
717	DYB=YY(IJW)-YY(IMJW)	
718	XB=HAF*(XX(IJW)+XX(IMJW))	
719	YB=HAF*(YY(IJW)+YY(IMJW))	
720	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
721	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
722	DXBP=XBP-XB	
723	DYBP=YBP-YB	
724	DISN=DELTA(DXB, DYB, DXBP, DYBP)	
725	C--CHECK WEST BOUNDARY	
726	IF(JTBW(J).EQ.4) THEN	
727	IJW=J	
728	IMJW=IJW-1	
729	DXB=XX(IJW)-XX(IMJW)	
730	DYB=YY(IJW)-YY(IMJW)	
731	XB=HAF*(XX(IJW)+XX(IMJW))	
732	YB=HAF*(YY(IJW)+YY(IMJW))	
733	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
734	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
735	DXBP=XBP-XB	
736	DYBP=YBP-YB	
737	DISN=DELTA(DXB, DYB, DXBP, DYBP)	
738	ENDIF	
739	C--CHECK EAST BOUNDARY	
740	IF(JTBE(J).EQ.4) THEN	
741	IJW=IMNJ(IMJ)+J	
742	IMJW=IJW-1	
743	DXB=XX(IJW)-XX(IMJW)	
744	DYB=YY(IJW)-YY(IMJW)	
745	XB=HAF*(XX(IJW)+XX(IMJW))	
746	YB=HAF*(YY(IJW)+YY(IMJW))	
747	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
748	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
749	DXBP=XBP-XB	
750	DYBP=YBP-YB	
751	DISN=DELTA(DXB, DYB, DXBP, DYBP)	
752	ENDIF	
753	C	
754	DISN=MIN(DISN, DISNW, DISNE)	
755	RK=DISN*DEN(IJ)*SQRT(TE(IJ))/VISCOS	
756	IF(LAY2) THEN	
757	ALMU=C11*DISN*(1.0-EXP(-RK/ALMU))	
758	ALED=C11*DISN*(1.0-EXP(-RK/AED))	
759	IF(WOMG.NE.0.) THEN	
760	AME=1.0+1.30*(0.40*DUDY(IJ)-0.80*WOMG)*WOMG*	
761	*(TE(IJ)/(ED(IJ)+SMALL))**2	
762	AME=ABS(AME)	
763	ALMU=ALMU*AME**1.5	
764	ALED=ALED*AME**0.5	
765	END IF	
766	ED(IJ)=SQRT(TE(IJ))**3/ALED	
767	VIS2(IJ)=VISCOS*DEN(IJ)*CMU*SQRT(TE(IJ))*ALMU	
768	ELSE	
769	RT=DEN(IJ)*TE(IJ)*TE(IJ)/(VISCOS*ED(IJ))	
770	FMU(IJ)=(1.0+20.5/RT)*(1.0-EXP(-0.0165*RK))**2	
771	FLR1(IJ)=1.0*(0.05/FMU(IJ))**3	
772	FLR2(IJ)=1.0-EXP(-RT*RT)	
773	ENDIF	
774	CONTINUE	
775	CONTINUE	
776	C	
777	C...ALONG THE WEST BOUNDARY	
778	C	
779	C	
780	DO 75 J=2, NJM	
781	IF(JTBW(J).NE.4) GO TO 75	

Oct 12 1996 16:41	sskmod_2d	Page 12
782	DISNW= GREAT	
783	DISNS= GREAT	
784	DO 76 J=2, I2LN	
785	IJ=IMNJ(I)+J	
786	IJW=J	
787	IMJW=IJW-1	
788	DXB=XX(IJW)-XX(IMJW)	
789	DYB=YY(IJW)-YY(IMJW)	
790	XB=HAF*(XX(IJW)+XX(IMJW))	
791	YB=HAF*(YY(IJW)+YY(IMJW))	
792	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
793	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
794	DXBP=XBP-XB	
795	DYBP=YBP-YB	
796	DISN=DELTA(DXB, DYB, DXBP, DYBP)	
797	C--CHECK NORTH BOUNDARY	
798	IF(ITEN(I).EQ.4) THEN	
799	IJW=IMNJ(I)+NJM	
800	IMJW=IJW-NJ	
801	DXB=XX(IJW)-XX(IMJW)	
802	DYB=YY(IJW)-YY(IMJW)	
803	XB=HAF*(XX(IJW)+XX(IMJW))	
804	YB=HAF*(YY(IJW)+YY(IMJW))	
805	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
806	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
807	DXBP=XBP-XB	
808	DYBP=YBP-YB	
809	DISN=DELTA(DXB, DYB, DXBP, DYBP)	
810	ENDIF	
811	C--CHECK SOUTH BOUNDARY	
812	IF(ITBS(I).EQ.4) THEN	
813	IJW=IMNJ(I)+1	
814	IMJW=IJW-NJ	
815	DXB=XX(IJW)-XX(IMJW)	
816	DYB=YY(IJW)-YY(IMJW)	
817	XPW=HAF*(XX(IJW)+XX(IMJW))	
818	YPW=HAF*(YY(IJW)+YY(IMJW))	
819	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
820	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
821	DXBP=XBP-XPW	
822	DYBP=YBP-YPW	
823	DISN=DELTA(DXB, DYB, DXBP, DYBP)	
824	ENDIF	
825	C	
826	DISN=MIN(DISN, DISNS, DISNW)	
827	RK=DISN*DEN(IJ)*SQRT(TE(IJ))/VISCOS	
828	IF(LAY2) THEN	
829	ALMU=C11*DISN*(1.0-EXP(-RK/ALMU))	
830	ALED=C11*DISN*(1.0-EXP(-RK/AED))	
831	IF(WOMG.NE.0.) THEN	
832	AME=1.0+1.30*(0.40*DUDY(IJ)-0.80*WOMG)*WOMG*	
833	*(TE(IJ)/(ED(IJ)+SMALL))**2	
834	AME=ABS(AME)	
835	ALMU=ALMU*AME**1.5	
836	ALED=ALED*AME**0.5	
837	END IF	
838	ED(IJ)=SQRT(TE(IJ))**3/ALED	
839	VIS2(IJ)=VISCOS*DEN(IJ)*CMU*SQRT(TE(IJ))*ALMU	
840	ELSE	
841	RT=DEN(IJ)*TE(IJ)*TE(IJ)/(VISCOS*ED(IJ))	
842	FMU(IJ)=(1.0+20.5/RT)*(1.0-EXP(-0.0165*RK))**2	
843	FLR1(IJ)=1.0*(0.05/FMU(IJ))**3	
844	FLR2(IJ)=1.0-EXP(-RT*RT)	
845	ENDIF	
846	CONTINUE	
847	CONTINUE	
848	76	
849	C	
850	C...ALONG THE EAST BOUNDARY	
851	C	
852	DO 78 J=2, NJM	

Oct 12 1996 16:41	sskmod_2d	Page 13
853	IF (JTBE(IJ, NE.4) .EQ. 4) GO TO 78	
854	DISN=GREAT	
855	DISNN=GREAT	
856	DISNS=GREAT	
857	DO 79 I=I2LE, NIM	
858	IJ=IMNJ(I)+J	
859	IJW=IMNJ(NIM)+J	
860	IMJW=IJW-1	
861	DXB=XX(IJW) -XX(IMJW)	
862	DYB=YY(IJW) -YY(IMJW)	
863	XB=HAF*(XX(IJW)+XX(IMJW))	
864	YB=HAF*(YY(IJW)+YY(IMJW))	
865	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
866	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
867	DXBP=XBP-XB	
868	DYBP=YBP-YB	
869	DISN=DELTA(DXB, DYB, DXBP, DYBP)	
870	C--CHECK NORTH BOUNDARY	
871	IF (ITBN(I) .EQ. 4) THEN	
872	IJW=IMNJ(I)+NJM	
873	IMJW=IJW-NJ	
874	DXB=XX(IJW) -XX(IMJW)	
875	DYB=YY(IJW) -YY(IMJW)	
876	XB=HAF*(XX(IJW)+XX(IMJW))	
877	YB=HAF*(YY(IJW)+YY(IMJW))	
878	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
879	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
880	DXBP=XBP-XB	
881	DYBP=YBP-YB	
882	DISN=DELTA(DXB, DYB, DXBP, DYBP)	
883	ENDIF	
884	C--CHECK SOUTH BOUNDARY	
885	IF (ITBS(I) .EQ. 4) THEN	
886	IJW=IMNJ(I)+1	
887	IMJW=IJW-NJ	
888	DXB=XX(IJW) -XX(IMJW)	
889	DYB=YY(IJW) -YY(IMJW)	
890	XB=HAF*(XX(IJW)+XX(IMJW))	
891	YB=HAF*(YY(IJW)+YY(IMJW))	
892	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
893	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
894	DXBP=XBP-XB	
895	DYBP=YBP-YB	
896	DISN=DELTA(DXB, DYB, DXBP, DYBP)	
897	ENDIF	
898	C	
899	DISN=MIN(DISN, DISNS, DISNN)	
900	RK=DISN*DEN(IJ)*SQRT(TE(IJ))/VISCOS	
901	IF (LAY2) THEN	
902	ALMU=C11*DISN*(1.0-EXP(-RK/ALMU))	
903	ALSD=C11*DISN*(1.0-EXP(-RK/AED))	
904	IF (WONG, NE.0.) THEN	
905	AME=1.0+1.30*(0.40*DUDY(IJ)-0.80*WONG)*WONG*	
906	*(TE(IJ)/(ED(IJ)*SMALL))**2	
907	AME=ABS(AME)	
908	ALMU=ALMU*AME**1.5	
909	ALED=ALED*AME**0.5	
910	END IF	
911	ED(IJ)=SQRT(TE(IJ))**3/ALED	
912	VIS2(IJ)=VISCOS*DEN(IJ)*CMU*SQRT(TE(IJ))*ALMU	
913	ELSE	
914	RT=DEN(IJ)*TE(IJ)*TE(IJ)/(VISCOS*ED(IJ))	
915	FMD(IJ)=(1.0+20.5/RT)*(1.0-EXP(-0.0165*RT))**2	
916	FLR1(IJ)=1.0*(0.05/FMD(IJ))**3	
917	FLR2(IJ)=1.0-EXP(-RT*RT)	
918	ENDIF	
919	CONTINUE	
920	CONTINUE	
921	C	
922	RETURN	
923	END	

Oct 12 1996 16:41	sskmod_2d	Page 14
924	C	
925	FUNCTION DELTA(DXB, DYB, DXBP, DYBP)	
926	ARW=SQRT(DXB**2+DYB**2)	
927	DXB=DXB/ARW	
928	DYB=DYB/ARW	
929	DELTA=DXB*DXBP+DYB*DYBP	
930	DELTA=SQRT(DXB**2+DYB**2-DELTA**2)	
931	RETURN	
932	END	
933	C	
934	C--SUBROUTINE SOLSIP(PHI, IPHI)	
935	-----	
936	-----	
937	INCLUDE 'kmod.h'	
938	C	
939	DIMENSION PHI(NXNY)	
940	ID=IPHI	
941	L=0	
942	DO 5 I=2, NIM	
943	DO 5 J=2, NJM	
944	IQ=IMNJ(I)+J	
945	API=1.0/AP(IQ)	
946	AP(IQ)=1.0	
947	AE(IQ)=AE(IQ)*API	
948	AW(IQ)=AW(IQ)*API	
949	AN(IQ)=AN(IQ)*API	
950	AS(IQ)=AS(IQ)*API	
951	SU(IQ)=SU(IQ)*API	
952	CONTINUE	
953	C	
954	DO 10 I=2, NIM	
955	DO 11 J=2, NJM	
956	IJ=IMNJ(I)+J	
957	IJW=IJ-1	
958	IMJ=IJ-NJ	
959	BW(IJ)=-AW(IJ)/(1.+ALFA*BN(IMJ))	
960	BS(IJ)=-AS(IJ)/(1.+ALFA*BE(IJM))	
961	POM1=ALFA*BW(IJ)*BN(IMJ)	
962	POM2=ALFA*BS(IJ)*BE(IJM)	
963	BP(IJ)=AP(IJ)+POM1+POM2-BW(IJ)*BE(IMJ)-BS(IJ)*BN(IJM)	
964	BN(IJ)=(-AN(IJ)-POM1)/(BP(IJ)+SMALL)	
965	BE(IJ)=(-AE(IJ)-POM2)/(BP(IJ)+SMALL)	
966	11 CONTINUE	
967	10 CONTINUE	
968	100 NSTP=NSWP(ID)	
969	CONTINUE	
970	L=L+1	
971	RESORP=0.	
972	DO 20 I=2, NIM	
973	DO 21 J=2, NJM	
974	IJ=IMNJ(I)+J	
975	RES(IJ)=AN(IJ)*PHI(IJ+1)+AS(IJ)*PHI(IJ-1)+AE(IJ)*PHI(IJ+NJ)+	
976	AW(IJ)*PHI(IJ-NJ)+SU(IJ)-AP(IJ)*PHI(IJ)	
977	RESORP=RES(IJ)+RES(IJ)	
978	RES(IJ)=(RES(IJ)-BS(IJ)*RES(IJ-1)-BW(IJ)*RES(IJ-NJ))/	
979	BP(IJ)+SMALL)	
980	21 CONTINUE	
981	20 CONTINUE	
982	IF (L.EQ.1) RESOR(ID)=RESORP	
983	RSM=SOR(ID)*RESOR(ID)	
984	DO 30 I=2, NIM	
985	II=NIM+2-I	
986	DO 31 J=2, NJM	
987	JJ=NJM+2-J	
988	IJ=IMNJ(II)+JJ	
989	RES(IJ)=RES(IJ)-BN(IJ)*RES(IJ+1)-BE(IJ)*RES(IJ+NJ)	
990	PHI(IJ)=PHI(IJ)+RES(IJ)	
991	31 CONTINUE	
992	30 CONTINUE	
993	IF (TEST) PRINT 1, L, RESORP	
994	IF (RESORP.GT.RSM.AND.L.LT.NSTP) GO TO 100	

Oct 12 1996 16:41 sskemod_2d Page 15

```

995 IF (RESORP.GT.RSM.AND.L.GE.NSTP) WRITE(6,2)
996 1 FORMAT(10X,15,' SWEEP, RESOR =',E12.4)
997 2 FORMAT(//,10X,' SIP SOL DID NOT CONVERGE ',//)
998 RETURN
999 END
1000 C
1001 C-----
1002 SUBROUTINE USERM
1003 C-----
1004 INCLUDE 'kemod.h'
1005 C
1006 C
1007 C
1008 C
1009 C
1010 C
1011 C-----
1012 DO 80 I=1,N1
1013 DO 80 J=1,NJ
1014 IJ=IMNJ(I)+J
1015 VISOLD=VIS(IJ)
1016 VIS(IJ)=VISCOS
1017 IF (ED(IJ).GT.SMALL)
1018 & VIS(IJ)=FNU(IJ)*DEN(IJ)*TE(IJ)**2*CMU/ED(IJ)+VISCOS
1019 VIS(IJ)=URFVIS*VIS(IJ)+(1.-URFVIS)*VISOLD
1020 80 CONTINUE
1021 C
1022 IF (LAY2) THEN
1023 DO 81 I=2,NIM
1024 IF (ITBS(I).NE.4) GO TO 82
1025 DO 83 J=2,J2LS
1026 IJ=IMNJ(I)+J
1027 VIS(IJ)=VIS2(IJ)
1028 83 CONTINUE
1029 82 IF (ITEN(I).NE.4) GO TO 81
1030 DO 84 J=J2LN,NJM
1031 IJ=IMNJ(I)+J
1032 VIS(IJ)=VIS2(IJ)
1033 84 CONTINUE
1034 81 CONTINUE
1035 DO 85 J=2,NJM
1036 IF (JTB(I).NE.4) GO TO 86
1037 DO 87 I=I2LE,NIM
1038 IJ=IMNJ(I)+J
1039 VIS(IJ)=VIS2(IJ)
1040 87 CONTINUE
1041 86 IF (JTBW(J).NE.4) GO TO 85
1042 DO 88 I=2,I2LM
1043 IJ=IMNJ(I)+J
1044 VIS(IJ)=VIS2(IJ)
1045 88 CONTINUE
1046 85 CONTINUE
1047 C
1048 ENDIF
1049 C
1050 RETURN
1051 C-----
1052 ENTRY MODPHI
1053 C-----
1054 GO TO (800,900) IDIR
1055 C
1056 C-----BOUNDARY CONDITIONS FOR KINETIC TURBULENT ENERGY
1057 C
1058 C
1059 800 CONTINUE
1060 C-----SOUTH BOUNDARY
1061 DO 810 I=2,NIM
1062 IJ=IMNJ(I)+2
1063 GO TO (811,812,813,814) ITBS(I)
1064 811 CONTINUE
1065 SU(IJ)=SU(IJ)+AS(IJ)*TE(IJ-1)

```

Oct 12 1996 16:41 sskemod_2d Page 16

```

1066 BP(IJ)=BP(IJ)+AS(IJ)
1067 GO TO 815
1068 812 TE(IJ-1)=TE(IJ)
1069 GO TO 815
1070 813 CONTINUE
1071 IJ=IJ-1
1072 IJ=IJ+NJ
1073 IMJ=IJ-NJ
1074 FXE1=FX(IJ)
1075 FXE2=FX(IMJ)
1076 FXW1=1.-FXE1
1077 FXW2=1.-FXE2
1078 DXB=XX(IJ)-XX(IMJ)
1079 DYB=YY(IJ)-YY(IMJ)
1080 DXBP=QTR*(XX(IJ+1)-XX(IJ)+XX(IMJ+1)-XX(IMJ))
1081 DYBP=QTR*(YY(IJ+1)-YY(IJ)+YY(IMJ+1)-YY(IMJ))
1082 FAL=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
1083 DEL=TE(IJ)*(FXW1-FXE2)+TE(IPJ)*FXE1-TE(IMJ)*FXW2
1084 TE(IJ)=TE(IJ+1)-DEL*FAC
1085 IJ=IJ+1
1086 GO TO 815
1087 814 CONTINUE
1088 IF (.NOT. LAY2 .AND. .NOT. LRE) GEN(IJ)=CENTS(I)
1089 RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))
1090 IF (AKSI) THEN
1091 IMJ=IJ-NJ
1092 IJM=IJ-1
1093 IJP=IJ+1
1094 IRJ=IJ+NJ
1095 FYN=FY(IJ)
1096 FYS=1.0-FYN
1097 FXE=FX(IJ)
1098 FXW=1.0-FXE
1099 DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
1100 DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))
1101 DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
1102 DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))
1103 WN=W(IJP)*FYN+W(IJ)*FYS
1104 WS=W(IPJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))
1105 WE=W(IJ)*FXE+W(IJ)*FXW
1106 WM=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))
1107 DWEN=WN-WS
1108 DMNS=WN-WS
1109 GEN(IJ)=GEN(IJ)+((DYNS*DWEN-DYEW*DMNS)**2+
1110 & (DXEW*DMNS-DXNS*DWEN*(W(IJ)/RP)*ARE(IJ))**2)
1111 & / (ARE(IJ)**2)
1112 & +2.*(V(IJ)/RP)**2 )*(VIS(IJ)-VISCOS)
1113 ENDIF
1114 SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
1115 815 CONTINUE
1116 AS(IJ)=0.0
1117 810 CONTINUE
1118 C
1119 C-----NORTH BOUNDARY
1120 DO 820 I=2,NIM
1121 IJ=IMNJ(I)+NJM
1122 GO TO (821,822,823,824) ITBN(I)
1123 821 CONTINUE
1124 SU(IJ)=SU(IJ)+AN(IJ)*TE(IJ+1)
1125 BP(IJ)=BP(IJ)+AN(IJ)
1126 GO TO 825
1127 822 TE(IJ+1)=TE(IJ)
1128 GO TO 825
1129 823 CONTINUE
1130 IJ=IJ+1
1131 IPJ=IJ+NJ
1132 IMJ=IJ-NJ
1133 FXE1=FX(IJ)
1134 FXE2=FX(IMJ)
1135 FXW1=1.-FXE1
1136 FXW2=1.-FXE2

```

Oct 12 1996 16:41	sskmod_2d	Page 18
1208	IMJ=IJ-NJ	
1209	IJM=IJ-1	
1210	IJP=IJ+1	
1211	IPJ=IJ+NJ	
1212	FYN=FY(IJ)	
1213	FYS=1.0-FYN	
1214	FXE=FX(IJ)	
1215	FXW=1.0-FXE	
1216	DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))	
1217	DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))	
1218	DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))	
1219	DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IJM)-YY(IMJ-1))	
1220	WN=W(IJP)*FYN+W(IJ)*FYS	
1221	WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))	
1222	WE=W(IJP)*FXE+W(IJ)*FXW	
1223	WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))	
1224	DWEW=WE-WN	
1225	DWNS=WN-WS	
1226	GEN(IJ)=GEN(IJ)+((DYNS*DWEW-DYEW*DWNS)**2+	
1227	& (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)	
1228	& / (ARE(IJ)**2)	
1229	& +2.*(V(IJ)/RP)**2)*(VIS(IJ)-VISCOS)	
1230	ENDIF	
1231	SU(IJ)=APV(IJ)*GEN(IJ)*VOL(IJ)	
1232	835 CONTINUE	
1233	AW(IJ)=0	
1234	830 CONTINUE	
1235	C-----EAST BOUNDARY	
1236	DO 840 J=2,NJM	
1237	IJ=IMNJ(NJM)+J	
1238	GO TO (841,842,843,844) JTBE(J)	
1239	841 CONTINUE	
1240	SU(IJ)=SU(IJ)+AE(IJ)*TE(IJ+NJ)	
1241	BP(IJ)=BP(IJ)+AE(IJ)	
1242	GO TO 845	
1243	842 TE(IJ+NJ)=TE(IJ)	
1244	GO TO 845	
1245	843 CONTINUE	
1246	IJ=IJ+NJ	
1247	IJP=IJ+1	
1248	IJM=IJ-1	
1249	FYN1=FY(IJ)	
1250	FYN2=FY(IJM)	
1251	FYS1=1.-FYN1	
1252	FYS2=1.-FYN2	
1253	DXB=XX(IJ)-XX(IJM)	
1254	DYB=YY(IJ)-YY(IJM)	
1255	DXBP=QTR*(XX(IJ-NJ-NJ)-XX(IJ)+XX(IJM-NJ-NJ)-XX(IJM))	
1256	DYBP=QTR*(YY(IJ-NJ-NJ)-YY(IJ)+YY(IJM-NJ-NJ)-YY(IJM))	
1257	FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)	
1258	DEL=TE(IJ)*(FYS1-FYN2)+TE(IJP)*FYN1-TE(IJM)*FYS2	
1259	TE(IJ)=TE(IJ-NJ)-DEL*FAC	
1260	IJ=IJ-NJ	
1261	GO TO 845	
1262	844 CONTINUE	
1263	IF(.NOT. LAY2 .AND. .NOT. LRE) GEN(IJ)=GENTEE(J)	
1264	RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))	
1265	IF(AKSI) THEN	
1266	IMJ=IJ-NJ	
1267	IJM=IJ-1	
1268	IJP=IJ+1	
1269	IPJ=IJ+NJ	
1270	FYN=FY(IJ)	
1271	FYS=1.0-FYN	
1272	FXE=FX(IJ)	
1273	FXW=1.0-FXE	
1274	DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))	
1275	DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))	
1276	DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))	
1277	DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IJM)-YY(IMJ-1))	
1278	WN=W(IJP)*FYN+W(IJ)*FYS	

Oct 12 1996 16:41	sskmod_2d	Page 17
1137	DXB=XX(IJ)-XX(IMJ)	
1138	DYB=YY(IJ)-YY(IMJ)	
1139	DXBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))	
1140	DYBP=QTR*(YY(IJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))	
1141	FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)	
1142	DEL=TE(IJ)*(FXW1-FXE2)+TE(IJP)*FYN1-TE(IMJ)*FXW2	
1143	TE(IJ)=TE(IJ-1)-DEL*FAC	
1144	IJ=IJ-1	
1145	GO TO 825	
1146	824 CONTINUE	
1147	IF(.NOT. LAY2 .AND. .NOT. LRE) GEN(IJ)=GENTN(I)	
1148	RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))	
1149	IF(AKSI) THEN	
1150	IMJ=IJ-NJ	
1151	IJM=IJ-1	
1152	IJP=IJ+1	
1153	IPJ=IJ+NJ	
1154	FYN=FY(IJ)	
1155	FYS=1.0-FYN	
1156	FXE=FX(IJ)	
1157	FXW=1.0-FXE	
1158	DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))	
1159	DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))	
1160	DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))	
1161	DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IJM)-YY(IMJ-1))	
1162	WN=W(IJP)*FYN+W(IJ)*FYS	
1163	WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))	
1164	WE=W(IJP)*FXE+W(IJ)*FXW	
1165	WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))	
1166	DWEW=WE-WN	
1167	DWNS=WN-WS	
1168	GEN(IJ)=GEN(IJ)+((DYNS*DWEW-DYEW*DWNS)**2+	
1169	& (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)	
1170	& / (ARE(IJ)**2)	
1171	& +2.*(V(IJ)/RP)**2)*(VIS(IJ)-VISCOS)	
1172	ENDIF	
1173	SU(IJ)=APV(IJ)*GEN(IJ)*VOL(IJ)	
1174	825 CONTINUE	
1175	AW(IJ)=0	
1176	820 CONTINUE	
1177	C-----WEST BOUNDARY	
1178	DO 830 J=2,NJM	
1179	IJ=IMNJ(2)+J	
1180	GO TO (831,832,833,834) JTBW(J)	
1181	831 CONTINUE	
1182	SU(IJ)=SU(IJ)+AW(IJ)*TE(IJ-NJ)	
1183	BP(IJ)=BP(IJ)+AW(IJ)	
1184	GO TO 835	
1185	832 TE(IJ-NJ)=TE(IJ)	
1186	GO TO 835	
1187	833 CONTINUE	
1188	IJ=J	
1189	IJP=IJ+1	
1190	IJM=IJ-1	
1191	FYN1=FY(IJ)	
1192	FYN2=FY(IJM)	
1193	FYS1=1.-FYN1	
1194	FYS2=1.-FYN2	
1195	DXB=XX(IJ)-XX(IJM)	
1196	DYB=YY(IJ)-YY(IJM)	
1197	DXBP=QTR*(XX(IJ+NJ)-XX(IJ)+XX(IJM+NJ)-XX(IJM))	
1198	DYBP=QTR*(YY(IJ+NJ)-YY(IJ)+YY(IJM+NJ)-YY(IJM))	
1199	FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)	
1200	DEL=TE(IJ)*(FYS1-FYN2)+TE(IJP)*FYN1-TE(IJM)*FYS2	
1201	TE(IJ)=TE(IJ+NJ)-DEL*FAC	
1202	IJ=IJ+NJ	
1203	GO TO 835	
1204	834 CONTINUE	
1205	IF(.NOT. LAY2 .AND. .NOT. LRE) GEN(IJ)=GENTW(J)	
1206	RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))	
1207	IF(AKSI) THEN	

Oct 12 1996 16:41	sskmod_2d	Page 19
1279	WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))	
1280	WE=W(IPJ)*FYE+W(IJ)*FXW	
1281	WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))	
1282	DWEW=WE-W	
1283	DWNS=WN-WS	
1284	GEN(IJ)=GEN(IJ)+((DWNS*DWEW-DYEW*DWNS)**2+ (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2) /(ARE(IJ)**2)	
1285	&	
1286	& +2.*(V(IJ)/RP)**2)*(VIS(IJ)-VISCOS)	
1287	&	
1288	ENDIF	
1289	SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)	
1290	845 CONTINUE	
1291	AE(IJ)=0.0	
1292	840 CONTINUE	
1293	C	
1294	RETURN	
1295	C	
1296	C-----BOUNDARY CONDITIONS FOR DISSIPATION OF KIN. TURB. ENERGY	
1297	C	
1298	900 CONTINUE	
1299	C	
1300	CMU25=SQRT(SORT(CMU))	
1301	CMU75=CMU25**3	
1302	C	
1303	C-----SOUTH BOUNDARY	
1304	DO 910 I=2,NJM	
1305	IJ=IMNJ(I)+2	
1306	GO TO (911,912,913,914) ITBS(I)	
1307	911 CONTINUE	
1308	SU(IJ)=SU(IJ)+AS(IJ)*ED(IJ-1)	
1309	BP(IJ)=BP(IJ)+AS(IJ)	
1310	GO TO 915	
1311	912 ED(IJ-NJ)=ED(IJ)	
1312	GO TO 915	
1313	913 CONTINUE	
1314	IJ=IJ-1	
1315	IPJ=IG+NJ	
1316	IMJ=IG-NJ	
1317	FXE1=FX(IJ)	
1318	FXE2=FX(IMJ)	
1319	FXW1=1.-FXE1	
1320	FXW2=1.-FXE2	
1321	DXB=XX(IJ)-XX(IMJ)	
1322	DYB=YY(IJ)-YY(IMJ)	
1323	DYBP=QTR*(XX(IJ+1)-XX(IMJ+1)-XX(IJ)+XX(IMJ+1)-XX(IMJ))	
1324	DYBP=QTR*(YY(IJ+1)-YY(IMJ+1)-YY(IJ)+YY(IMJ+1)-YY(IMJ))	
1325	FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)	
1326	DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2	
1327	ED(IJ)=ED(IJ+1)-DEL*FAC	
1328	IJ=IJ+1	
1329	GO TO 915	
1330	914 CONTINUE	
1331	IF(LRE) THEN	
1332	IJ=IJ-1	
1333	IPJ=IG+NJ	
1334	IMJ=IG-NJ	
1335	FXE1=FX(IJ)	
1336	FXE2=FX(IMJ)	
1337	FXW1=1.-FXE1	
1338	FXW2=1.-FXE2	
1339	DXB=XX(IJ)-XX(IMJ)	
1340	DYB=YY(IJ)-YY(IMJ)	
1341	DXBP=QTR*(XX(IJ+1)-XX(IMJ+1)-XX(IJ)+XX(IMJ+1)-XX(IMJ))	
1342	DYBP=QTR*(YY(IJ+1)-YY(IMJ+1)-YY(IJ)+YY(IMJ+1)-YY(IMJ))	
1343	FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)	
1344	DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2	
1345	ED(IJ)=ED(IJ+1)-DEL*FAC	
1346	IJ=IJ+1	
1347	ELSE	
1348	C	
1349	TE(IJ)=ABS(TE(IJ))	

Oct 12 1996 16:41	sskmod_2d	Page 20
1350	SU(IJ)=CMU75*TE(IJ)*SORT(TE(IJ))/(CAPPA*DNS(I))*GREAT	
1351	BP(IJ)=GREAT	
1352	ENDIF	
1353	915 CONTINUE	
1354	AS(IJ)=0.0	
1355	910 CONTINUE	
1356	C-----NORTH BOUNDARY	
1357	DO 920 I=2,NJM	
1358	IJ=IMNJ(I)+NJM	
1359	GO TO (921,922,923,924) ITBN(I)	
1360	921 CONTINUE	
1361	SU(IJ)=SU(IJ)+AN(IJ)*ED(IJ+1)	
1362	BP(IJ)=BP(IJ)+AN(IJ)	
1363	GO TO 925	
1364	922 ED(IJ+1)=ED(IJ)	
1365	GO TO 925	
1366	923 CONTINUE	
1367	IJ=IJ+1	
1368	IPJ=IG+NJ	
1369	IMJ=IG-NJ	
1370	FXE1=FX(IJ)	
1371	FXE2=FX(IMJ)	
1372	FXW1=1.-FXE1	
1373	FXW2=1.-FXE2	
1374	DXB=XX(IJ)-XX(IMJ)	
1375	DYB=YY(IJ)-YY(IMJ)	
1376	DYBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))	
1377	DYBP=QTR*(YY(IJ-2)-YY(IMJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))	
1378	FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)	
1379	DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2	
1380	ED(IJ)=ED(IJ-1)-DEL*FAC	
1381	IJ=IJ-1	
1382	GO TO 925	
1383	924 CONTINUE	
1384	IF(LRE) THEN	
1385	IJ=IJ+1	
1386	IPJ=IG+NJ	
1387	IMJ=IG-NJ	
1388	FXE1=FX(IJ)	
1389	FXE2=FX(IMJ)	
1390	FXW1=1.-FXE1	
1391	FXW2=1.-FXE2	
1392	DXB=XX(IJ)-XX(IMJ)	
1393	DYB=YY(IJ)-YY(IMJ)	
1394	DYBP=QTR*(XX(IJ-2)-XX(IMJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))	
1395	DYBP=QTR*(YY(IJ-2)-YY(IMJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))	
1396	FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)	
1397	DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2	
1398	ED(IJ)=ED(IJ-1)-DEL*FAC	
1399	IJ=IJ-1	
1400	ELSE	
1401	TE(IJ)=ABS(TE(IJ))	
1402	SU(IJ)=CMU75*TE(IJ)*SORT(TE(IJ))/(CAPPA*DNN(I))*GREAT	
1403	BP(IJ)=GREAT	
1404	ENDIF	
1405	925 CONTINUE	
1406	AN(IJ)=0.0	
1407	920 CONTINUE	
1408	C-----WEST BOUNDARY	
1409	DO 930 J=2,NJM	
1410	IJ=IMNJ(2)+J	
1411	GO TO (931,932,933,934) JTBW(J)	
1412	931 CONTINUE	
1413	SU(IJ)=SU(IJ)+AW(IJ)*ED(IJ-NJ)	
1414	BP(IJ)=BP(IJ)+AW(IJ)	
1415	GO TO 935	
1416	932 ED(IJ-NJ)=ED(IJ)	
1417	GO TO 935	
1418	933 CONTINUE	
1419	IJ=J	
1420	IJP=IJ+1	

```

1492 FYN1=FY(IJ)
1493 FYN2=FY(IJM)
1494 FYS1=1.-FYN1
1495 FYS2=1.-FYN2
1496 DXB=XX(IJ)-XX(IJM)
1497 DYB=YY(IJ)-YY(IJM)
1498 DXBP=QTR*(XX(IJ-NJ)-XX(IJ-NJ-NJ)-XX(IJ-NJ)-XX(IJM))
1499 DYBP=QTR*(YY(IJ-NJ)-YY(IJ)-YY(IJ-NJ-NJ)-YY(IJM))
1500 FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
1501 DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
1502 ED(IJ)=ED(IJ-NJ)-DEL*FAC
1503 IJ=IJ-NJ
1504 ELSE
1505 TE(IJ)=ABS(TE(IJ))
1506 SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNE(J))*GREAT
1507 BP(IJ)=GREAT
1508 ENDIF
1509 945 CONTINUE
1510 AE(IJ)=0.0
1511 940 CONTINUE
1512 C
1513 C
1514 C
1515 C--- SUBROUTINE 'MODIFY' TO MODIFY SHEAR STRESSES IN
1516 C--- NEAR WALL GRID POINT
1517 C
1518 C--- SUBROUTINE MODIFY (SUR, BPR)
1519 C
1520 INCLUDE 'kmod.h'
1521 DIMENSION SUR(NXNY), BPR(NXNY)
1522 C
1523 DATA CMU25,CAPPA,ELOG/0.5477,0.4197,9.0/
1524 C
1525 DO 10 I=1,NJ
1526 DO 10 J=1,NJ
1527 IJ=IMNJ(I)+J
1528 SU(IJ)=SUR(IJ)
1529 BP(IJ)=BPR(IJ)
1530 CONTINUE
1531 C
1532 C--- CHECK WALL SOUTH BOUNDARY
1533 C
1534 DO 600 I=2,NJM
1535 IJ=IMNJ(I)+2
1536 IF(ITBS(I).EQ.4) THEN
1537 LB=IJ
1538 LW=IJ-1
1539 TEPR=SQRT(TE(IJ))
1540 DELN=DNS(I)
1541 DXB=XX(IJ-1)-XX(IJ-NJ-1)
1542 DYB=YY(IJ-1)-YY(IJ-NJ-1)
1543 RB=HAF*(R(IJ-1)+R(IJ-NJ-1))
1544 DEN=DEN(IJ)
1545 CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,
1546 & TAU,SU,SUP,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)
1547 SU(IJ)=SU(IJ)+SUJ
1548 BP(IJ)=BP(IJ)+SUP
1549 SUVS(I)=SVU
1550 SPVS(I)=SVP
1551 SUMS(I)=SWU
1552 SPWS(I)=SWP
1553 GENTS(I)=GENTE
1554 IF(LRE) THEN
1555 IJ=IJ-1
1556 IFG=IG+NJ
1557 IMJ=IJ-NJ
1558 FXE1=FX(IJ)
1559 FXE2=FX(IMJ)
1560 FXW1=1.-FXE1
1561 FXW2=1.-FXE2
1562 DXB=XX(IJ)-XX(IMJ)

```

```

1421 IJM=IJ-1
1422 FYN1=FY(IJ)
1423 FYN2=FY(IJM)
1424 FYS1=1.-FYN1
1425 FYS2=1.-FYN2
1426 DXB=XX(IJ)-XX(IJM)
1427 DYB=YY(IJ)-YY(IJM)
1428 DXBP=QTR*(XX(IJ+NJ)-XX(IJ+NJ-NJ)-XX(IJ+NJ)-XX(IJM))
1429 DYBP=QTR*(YY(IJ+NJ)-YY(IJ)-YY(IJ+NJ-NJ)-YY(IJM))
1430 FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
1431 DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
1432 ED(IJ)=ED(IJ+NJ)-DEL*FAC
1433 IJ=IJ+NJ
1434 GO TO 935
1435 934 CONTINUE
1436 IF(LRE) THEN
1437 IJ=J
1438 IJP=IJ+1
1439 IJM=IJ-1
1440 FYN1=FY(IJ)
1441 FYN2=FY(IJM)
1442 FYS1=1.-FYN1
1443 FYS2=1.-FYN2
1444 DXB=XX(IJ)-XX(IJM)
1445 DXBP=QTR*(XX(IJ+NJ)-XX(IJ+NJ-NJ)-XX(IJ+NJ)-XX(IJM))
1446 DYBP=QTR*(YY(IJ+NJ)-YY(IJ)-YY(IJ+NJ-NJ)-YY(IJM))
1447 FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
1448 DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
1449 ED(IJ)=ED(IJ+NJ)-DEL*FAC
1450 IJ=IJ+NJ
1451 935 CONTINUE
1452 AW(IJ)=0.0
1453 TE(IJ)=ABS(TE(IJ))
1454 SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNE(J))*GREAT
1455 BP(IJ)=GREAT
1456 ENDIF
1457 935 CONTINUE
1458 AW(IJ)=0.0
1459 930 CONTINUE
1460 C-----EAST BOUNDARY
1461 DO 940 J=2,NJM
1462 IJ=IMNJ(NIM)+J
1463 GO TO (941,942,943,944) JTBE(J)
1464 941 CONTINUE
1465 SU(IJ)=SU(IJ)+AE(IJ)*ED(IJ+NJ)
1466 BP(IJ)=BP(IJ)+AE(IJ)
1467 GO TO 945
1468 942 ED(IJ+NJ)=ED(IJ)
1469 GO TO 945
1470 943 CONTINUE
1471 IJ=IJ+NJ
1472 IJP=IJ+1
1473 IJM=IJ-1
1474 FYN1=FY(IJ)
1475 FYN2=FY(IJM)
1476 FYS1=1.-FYN1
1477 FYS2=1.-FYN2
1478 DXB=XX(IJ)-XX(IJM)
1479 DYB=YY(IJ)-YY(IJM)
1480 DXBP=QTR*(XX(IJ-NJ)-XX(IJ-NJ-NJ)-XX(IJ-NJ)-XX(IJM))
1481 DYBP=QTR*(YY(IJ-NJ)-YY(IJ)-YY(IJ-NJ-NJ)-YY(IJM))
1482 FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
1483 DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2
1484 ED(IJ)=ED(IJ-NJ)-DEL*FAC
1485 IJ=IJ-NJ
1486 GO TO 945
1487 944 CONTINUE
1488 IF(LRE) THEN
1489 IJ=IJ+NJ
1490 IJP=IJ+1
1491 IJM=IJ-1

```


Oct 12 1996 16:41 **sskmod_2d** Page 25

```

1705 C----- SUBROUTINE 'WALLFN' TO SET WALL FUNCTIONS
1706 C
1707 C SUBROUTINE WALLFN (LB, LW, VISC, DENS, DXB, DYB, CMU25, ELOG, CAPPA,
1708 & TAU, SUU, SUP, SVU, SVP, SWU, SWP, GENTE, DELN, TEPR, RB)
1709
1710 C INCLUDE 'kmod.h'
1711 C
1712 C UP=U(LB)
1713 C VP=V(LB)
1714 C WP=W(LB)
1715 C UWALL=U(LW)
1716 C VWALL=V(LW)
1717 C WWALL=W(LW)
1718 C ARW=SQRT(DXB**2+DYB**2)
1719 C DXB=DXB/ARW
1720 C DYB=DYB/ARW
1721 C CONST=DENS*CMU25*TEPR
1722 C YPLS=DELN*CONST/VISC
1723 C TCOEF=VISC/DELN
1724 C IF (LAY2.OR.LRE) GOTO 10
1725 C IF (YPLS.LE.11.63) GO TO 10
1726 C UPLUS=LOG(ELOG*YPLS)/CAPPA
1727 C TCOEF=CONST/UPLUS
1728 C
1729 C 10 CONTINUE
1730 C VPINT=UP*DXB+VP*DYB
1731 C VPX=VPINT*DXB
1732 C VPY=VPINT*DYB
1733 C TAU=-TCOEF*(VPX*DXB+VPY*DYB)
1734 C ARW=ARW*RB
1735 C SUP=TCOEF*ARW*DXB**2
1736 C SVP=TCOEF*ARW*DYB**2
1737 C SWP=TCOEF*ARW
1738 C CON=TCOEF*ARW*DXB*DYB
1739 C SUU=-CON*VP*UWALL*TCOEF*ARW*DXB**2
1740 C SVU=-CON*UP*VWALL*TCOEF*ARW*DYB**2
1741 C SWU=TCOEF*ARW*WVALL*RB
1742 C
1743 C FOR MOVING WALL
1744 C
1745 C VPINT=VPINT*WP
1746 C VPINT=ABS(VPINT-SORT(UWALL*UWALL+VWALL*VWALL+WWALL*WWALL))
1747 C
1748 C GENTE=TCOEF*CONST*ABS(VPINT)/(CAPPA*DENS*DELN)
1749 C
1750 C RETURN
1751 C
1752 C -----
1753 C kmod.h
1754 C
1755 C COMMON BLOCK
1756 C
1757 C PARAMETER (NX=75)
1758 C PARAMETER (NY=50)
1759 C PARAMETER (NXNY=NX*NY)
1760 C COMMON/KEA1/URFK,URFE,URFVIS,NI,NJ,NIM,NJM,NINJ,IMNJ(NX),
1761 & IDIR,ITER,ITBS(NX),ITBN(NX),JTBW(NX),JTBE(NX),
1762 & TEST,AKSI,G
1763 C COMMON/KEA2/XX(NXNY),FY(NXNY),FX(NXNY),FY(NXNY),ARE(NXNY),
1764 & VOL(NXNY),DNS(NX),DNN(NX),DNW(NX),DNE(NX),R(NXNY)
1765 C COMMON/KEA3/AE(NXNY),AW(NXNY),AN(NXNY),AS(NXNY),AP(NXNY),
1766 & SU(NXNY)
1767 C COMMON/KEA4/APU(NXNY),APV(NXNY),BE(NXNY),BW(NXNY),BN(NXNY),
1768 & BS(NXNY),BP(NXNY),RES(NXNY),F1(NXNY),F2(NXNY)
1769 C COMMON/KEA5/U(NXNY),V(NXNY),W(NXNY),P(NXNY),GEN(NXNY),
1770 & TE(NXNY),ED(NXNY),DEN(NXNY),VIS(NXNY),RW(NXNY)
1771 C COMMON/KEA6/RESOR(2),PRTK,PRTE,URF(2),SORMAX,
1772 & VISCOS,C1,C2,CMU,ALFA,GREAT,SMALL,SNOW(NY),
1773 & UNW(NY),VWV(NY),PHINW(NY),HAF,QTR,DENSIT
1774 C COMMON/KEAA/NSWP(2),SOR(2)
1775 C COMMON/TURB/ FLR1(NXNY),FLR2(NXNY),VIS2(NXNY),PMU(NXNY),

```

Oct 12 1996 16:41 **sskmod_2d** Page 26

```

1776 & J2LS,J2LN,I2LE,I2LW,LRE,LAY2
1777 C COMMON/KEUVMOD/ SUVS(NX),SPVS(NX),SUVN(NX),SPVN(NX),
1778 & SUVW(NY),SPVW(NY),SUVE(NY),SPVE(NY),
1779 C & SPWN(NX),SPWS(NX),SPWE(NY),SPWV(NY),
1780 & SUWN(NX),SUWS(NX),SUME(NY),SUMW(NY),
1781 C & COMMON /KEGENER/ GENTS(NX),GENTN(NX),GENTW(NY),GENTEE(NY)
1782 C COMMON /ROT/ WONG,DUDY(NXNY),DVDX(NXNY)
1783 C LOGICAL TEST,AKSI,LRE,LAY2
1784
1785

```

CHAPTER 3

2D/Axisymmetric Multi-Time-Scale $k-\varepsilon$ Turbulence Model

Table of Contents

	page
3.1 Introduction	30
3.2 Theory and Model Equations	30
3.3 Module Evaluation	34
References	36
Figures	
Appendix B	37

In this section a description of the multi-time-scale k - ε turbulence model that is coded as a self contained computer program to compute turbulent flow quantities in two-dimensional or axisymmetric geometry is given. Detailed description of the module structure, variables used and how to interface the module with CFD flow solvers are given in Appendix B. The module has been tested as a separate self-contained unit using the REACT code [1] and was independently tested at the University of Alabama at Huntsville (UAH) using own code (MAST).

3.1 Introduction

Turbulent flows comprise fluctuating motions with a spectrum of sizes and time scales and different turbulent interactions are associated with different parts of the spectrum. In the single-time-scale turbulence models such as the k - ε turbulence model it is assumed that a single time scale (proportional to k/ε) can be used to describe the turbulent flow. In many complex flows turbulence is generally in spectral inequilibrium and a single time scale description is a simplification.

Figure 1, shows a sketch of a typical energy spectrum in a turbulent flow at high Reynolds number in a simplified split spectrum method. Two regions can be identified, the production range (at wave number $\kappa < \kappa_l$) where the kinetic energy (k_p) leaves this region at a rate (ε_p) and a high wave number or dissipation region ($\kappa > \kappa_l$) with kinetic energy (k_t) and energy dissipation rate (ε_t). Hanjalic et al. [2] developed a simple multiple-time-scale turbulence model based on a rational extension of the single scale equation ideas. In their model, a fixed ratio of the turbulent kinetic energy of large eddies (k_p) to that of the fine scale eddies (k_t) is used to partition the spectrum. Kim and Chen [3] improved on the simplified split spectrum by dynamically determining the location of the partition (i.e k_p/k_t) as part of the solution and is dependent on the turbulence intensity, production rate, energy transfer and dissipation rate. The variable partitioning method causes the effective eddy viscosity to decrease when production is high and to increase when production vanishes -a behavior consistent with experimental observations.

3.2 Theory and Model Equations

The multi-time-scale turbulence module is based on the variable partitioning of the turbulent energy spectrum proposed by Kim and Chen [3]. In this model the turbulent kinetic energy spectrum is divided into two sets of wave number regions giving two evolution equations for each region.

These equations represent the kinetic energy (k_p) and the energy transfer rate (ϵ_p) in the production range of the spectrum and the kinetic energy (k_t) and the energy dissipation rate (ϵ_t) in the dissipation range of the spectrum. This model allows the partition to move toward the high wave number region when production is high and toward the low wave number region when production vanishes.

The equations which describe the multi-time-scale turbulence model used are given below. The turbulent kinetic energy and the energy transfer rate equations for the energy containing large eddies are given as;

$$\rho \frac{Dk_p}{Dt} = \frac{\partial}{\partial x_i} \left[\left(\mu + \frac{\mu_t}{\sigma_{k_p}} \right) \frac{\partial k_p}{\partial x_i} \right] + G - \rho \epsilon_p \quad (1)$$

$$\rho \frac{D\epsilon_p}{Dt} = \frac{\partial}{\partial x_i} \left[\left(\mu + \frac{\mu_t}{\sigma_{\epsilon_p}} \right) \frac{\partial \epsilon_p}{\partial x_i} \right] + \frac{1}{\rho} C_{p1} \frac{G^2}{k_p} + C_{p2} \frac{G\epsilon_p}{k_p} - \rho C_{p3} \frac{\epsilon_p^2}{k_p} \quad (2)$$

where G is the turbulence production rate, given as

$$G = \mu_e \left\{ 2 \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial r} \right)^2 + \left(\frac{v}{r} \right)^2 \right] + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial w}{\partial r} - \frac{w}{r} \right)^2 \right\}$$

where μ is the viscosity

μ_t is the turbulent viscosity

k_p is the turbulent kinetic energy in the production range

ϵ_p is the energy transfer rate

σ_{k_p} and σ_{ϵ_p} are constants

C_{p1} , C_{p2} and C_{p3} are turbulent model constants

The turbulent kinetic energy and the dissipation rate equations for the high wave number small scale eddies region are given as;

$$\rho \frac{Dk_t}{Dt} = \frac{\partial}{\partial x_i} \left[\left(\mu + \frac{\mu_t}{\sigma_{k_t}} \right) \frac{\partial k_t}{\partial x_i} \right] + \rho \epsilon_p - \rho \epsilon_t \quad (3)$$

$$\rho \frac{D\varepsilon_t}{Dt} = \frac{\partial}{\partial x_i} \left[\left(\mu + \frac{\mu_t}{\sigma_{\varepsilon_t}} \right) \frac{\partial \varepsilon_t}{\partial x_i} \right] + \rho C_{t1} \frac{\varepsilon_p^2}{k_t} + \rho C_{t2} \frac{\varepsilon_t \varepsilon_p}{k_t} - \rho C_{t3} \frac{\varepsilon_t^2}{k_t} \quad (4)$$

where k_t is the turbulent kinetic energy in the dissipation range

ε_t is the energy dissipation rate

σ_{k_t} and σ_{ε_t} are constants

C_{t1} , C_{t2} and C_{t3} are turbulent model constants

The terms $\frac{1}{\rho} C_{P1} \frac{G^2}{k_p}$ and $\rho C_{t1} \frac{\varepsilon_p^2}{k_t}$ represent variable energy transfer functions. The first term increases the energy transfer rate when production is high and the second term increases the dissipation rate when the energy transfer rate is high. The turbulent viscosity is given as

$$\mu_t = \rho C_{\mu f} \frac{k^2}{\varepsilon_p} = \rho C_{\mu} \frac{k^2}{\varepsilon_t}$$

where $k = k_p + k_t$ is the total turbulent kinetic energy and $C_{\mu f}$ is a constant.

The model constants used are similar to those used by Kim and Chen [3]

$$\begin{aligned} \sigma_{k_p} &= 0.75, \quad \sigma_{\varepsilon_p} = 1.15, \quad \sigma_{k_t} = 0.75, \quad \sigma_{\varepsilon_t} = 1.15 \\ C_{P1} &= 0.21, \quad C_{P2} = 1.24, \quad C_{P3} = 1.84, \quad C_{t1} = 0.29 \\ C_{t2} &= 1.28, \quad C_{t3} = 1.66 \quad \text{and} \quad C_{\mu f} = 0.09 \end{aligned}$$

For turbulent flow analysis, equations (1)-(4) are solved by the module that is interfaced with a Reynolds averaged flow solver to compute the turbulent flow field. For an incompressible, steady and axisymmetric turbulent flow, a generalized equation that expresses the transport of turbulent flow can be written as;

$$\frac{\partial (\rho u \Phi)}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r} (\rho v r \Phi) = \frac{\partial}{\partial x} \left(\Gamma \Phi_x \frac{\partial \Phi}{\partial x} \right) + \frac{1}{r} \frac{\partial}{\partial r} \left(r \Gamma \Phi_r \frac{\partial \Phi}{\partial r} \right) + S_{\Phi} \quad (5)$$

where Φ is the dependent variable, which stands for $\Phi = u, v, w$ for the axial, radial and tangential velocities respectively. ρ is the fluid density, $\Gamma\Phi_x$ and $\Gamma\Phi_r$ are exchange coefficients in x and r -directions, respectively, and S_Φ is the source term for the variable Φ .

The source terms for the dependent variable are:

- Axial direction, $\Phi = u$, $\Gamma\Phi_x = 2\mu_e$, $\Gamma\Phi_r = \mu_e$ and

$$S_u = -\frac{\partial P}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r} (\mu_e r \frac{\partial v}{\partial x})$$

where μ_e is the eddy viscosity and P is the pressure

- Radial direction, $\Phi = v$, $\Gamma\Phi_x = \mu_e$, $\Gamma\Phi_r = 2\mu_e$ and

$$S_v = -\frac{\partial}{\partial x} \left(\mu_e \frac{\partial u}{\partial r} \right) - 2\mu_e \frac{v}{r^2} + \frac{\rho w^2}{r} - \frac{\partial P}{\partial r}$$

- Tangential direction, $\Phi = w$, $\Gamma\Phi_x = \mu_e$, $\Gamma\Phi_r = \mu_e$ and

$$S_w = -\frac{\rho v w}{r} - \frac{w}{r^2} \frac{\partial}{\partial r} (r \mu_e)$$

Equations (1)-(4) can also be written in a similar form as equation (5) where Φ stands for;

- Turbulent kinetic energy in the production range of the energy spectrum

$$\Phi = k_p, \quad \Gamma\Phi_x = \mu + \frac{\mu_t}{\sigma_{k_p}} = \Gamma\Phi_r \text{ and}$$

$$s_{k_p} = G - \rho \varepsilon_p$$

- Energy transfer rate in the production range of the energy spectrum

$$\Phi = \varepsilon_p, \quad \Gamma\Phi_x = \mu + \frac{\mu_t}{\sigma_{\varepsilon_p}} = \Gamma\Phi_r \text{ and}$$

$$s_{\varepsilon_p} = \frac{1}{\rho} C_{P1} \frac{G^2}{k_p} + C_{P2} \frac{G \varepsilon_p}{k_p} - \rho C_{P3} \frac{\varepsilon_p^2}{k_p}$$

- Turbulent kinetic energy in the dissipation range of the energy spectrum

$$\Phi = k_t, \quad \Gamma \Phi_x = \mu + \frac{\mu_t}{\sigma_{k_t}} = \Gamma \Phi_r \quad \text{and}$$

$$s_{k_t} = \rho \varepsilon_p - \rho \varepsilon_t$$

- Energy dissipation rate in the dissipation range of the energy spectrum

$$\Phi = \varepsilon_t, \quad \Gamma \Phi_x = \mu + \frac{\mu_t}{\sigma_{\varepsilon_t}} = \Gamma \Phi_r \quad \text{and}$$

$$s_{\varepsilon_t} = \rho C_{t1} \frac{\varepsilon_p^2}{k_t} + \rho C_{t2} \frac{\varepsilon_t \varepsilon_p}{k_t} - \rho C_{t3} \frac{\varepsilon_t^2}{k_t}$$

Near a wall, the wall function boundary conditions used are similar to that of Kim and Chen [3].

A two layer model for the multi-time-scale k - ε turbulence model similar to that of Chen and Patel [4] for the single-time-scale k - ε turbulence model is included in the present release.

3.3 Model Evaluation

The multi-time-scale k - ε module was evaluated by comparisons with experimental studies. One of the test problems considered was the backward facing step of Driver and Seegmiller [5] where the multi scale k - ε model predicted a recirculation length of 6.14 step heights (H) downstream of the step which is closer to the experimental value (6.10 H) than the standard k - ε model (5.35H).

The majority of the tests were conducted using Roback and Johnson's experimental data [6] for swirling confined double concentric jets. Preliminary analysis indicated some sensitivity to the ratio k_p / k_t at the inlet boundary, however, a value of 3 was found reasonable in the present analysis. Figures 2 and 3 show the streamline patterns for wall functions and two-layer near wall treatments respectively. The upper (a) and lower (b) parts correspond to the single-scale k - ε and the multi-scale k - ε models respectively. It can be seen from these contours that there are two recirculation zones in the chamber, one is near the expansion corner and another located in the central region and accurate predictions of this central region is very important in combusting swirling flows. Figures 4a and 4b, show the axial velocity along the centerline. In terms of strength and size of the central recirculation zone, the multi-scale k - ε model yields better agreement than the single-scale k - ε model. In the central recirculation region the k - ε model tends to connect the energy transfer rate to the local mean strain rate too strongly while the multi-scale model suppresses this tendency.

Figures 5 and 6 show the radial profiles of the mean axial velocity at different axial locations downstream of the inlet using the wall function and the two-layer near-wall treatments respectively. Similarly, figures 7 and 8 show the corresponding profiles for the tangential velocity, and figures 9 and 10 show the radial profiles of the axial normal turbulent intensity $(\overline{u'u'})^{1/2}$ using both the wall function and the two-layer near-wall treatments. In general, the numerical results indicate that the multi-scale model gives better agreement than the standard $k-\varepsilon$ model.

REFERENCES

1. Chon, J., Hadid, A. and Hamakiotes C. "REACT-2D Version 2.6 User's Manual" CFD Technology Center, Rocketdyne Division/Rockwell International, 1989.
2. Hanjalic, K., Launder, B. and Schiestel R. "Multiple-Time-Scale Concepts in Turbulent Transport Modeling" in Turbulent Shear Flows, Eds. Bradbury L. J. et al., vol. 2, Springer-Verlag, N.Y., pp. 36-49, 1980.
3. Kim, S.-W and Chen, C.-P "A Multiple-Time-Scale Turbulence Model Based on Variable Partitioning of the Turbulent Kinetic Energy Spectrum", Num. Heat Transfer, pt. B, vol. 16, no. 2, pp. 193-211, 1989.
4. Chen, H. C. and Patel, V. C. "Near-Wall Turbulence Models for Complex Flows Including Separation", AIAA Journal, vol. 26, no. 6, pp. 641-648, 1988.
5. Driver, D. and Seegmiller, H. "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow" AIAA J., vol. 23, pp. 163-171. 1985.
6. Roback, R. and Johnson, B "Mass and Momentum Turbulent Transport Experiment with Confined Swirling Co-axial Jets" NASA CR-168252, 1983.
7. Stone, H. "Iterative Solution of Implicit Approximations of Multi-Dimensional Partial Differential Equations", SIAM J. Num. Anal., vol. 5, pp 530-, 1968.

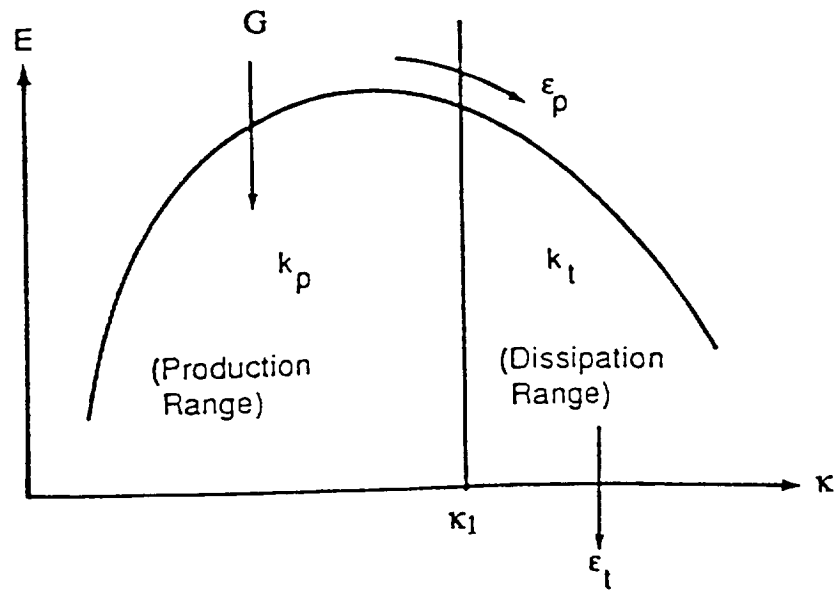
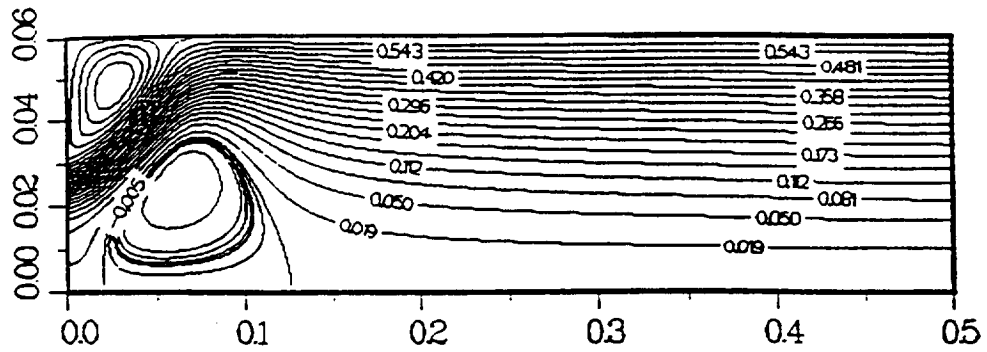


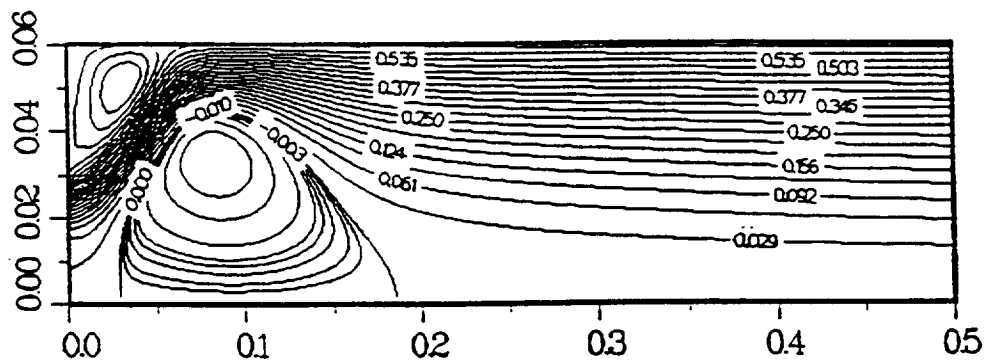
Figure 1. Description and nomenclature of the multiple-time-scale turbulence model;

$$k_p = \int_{\kappa=0}^{\kappa=\kappa_1} E \, d\kappa, \quad k_t = \int_{\kappa=\kappa_1}^{\kappa=\infty} E \, d\kappa$$

κ_1 = Partition wave number, E = Energy spectral density

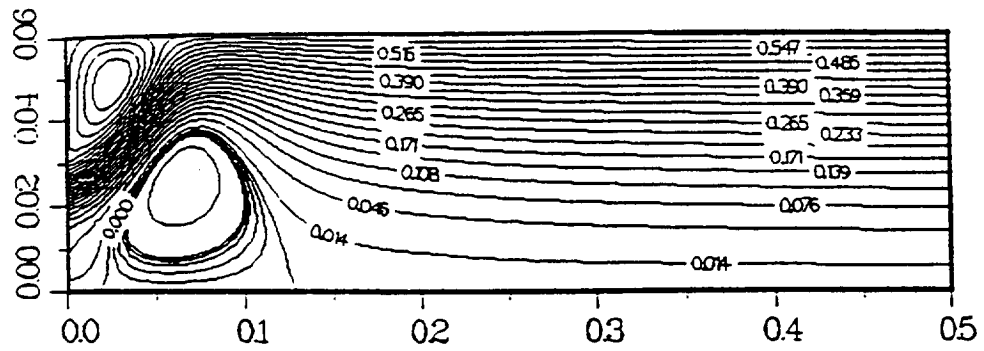


(a) $k-\epsilon$ model

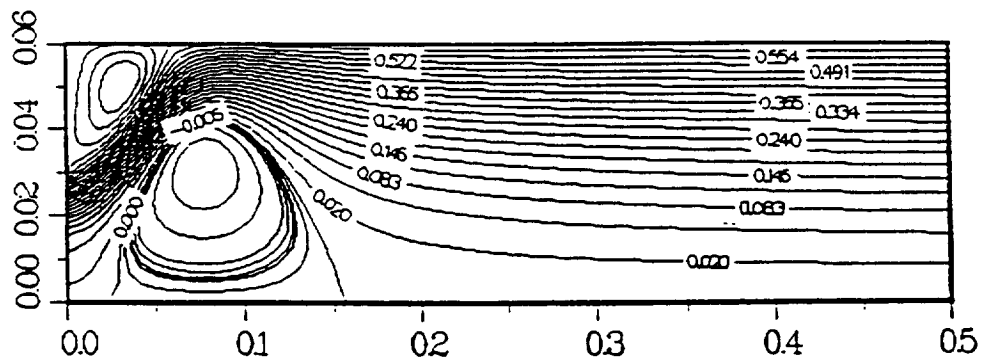


(b) M-S model

Figure 2. Stream-function contours of confined swirling jet flow using the wall function near wall treatment

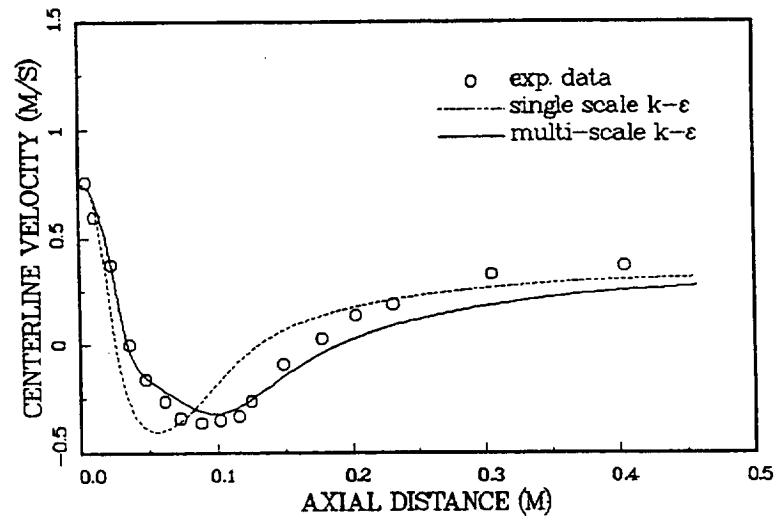


(a) $k-\epsilon$ model

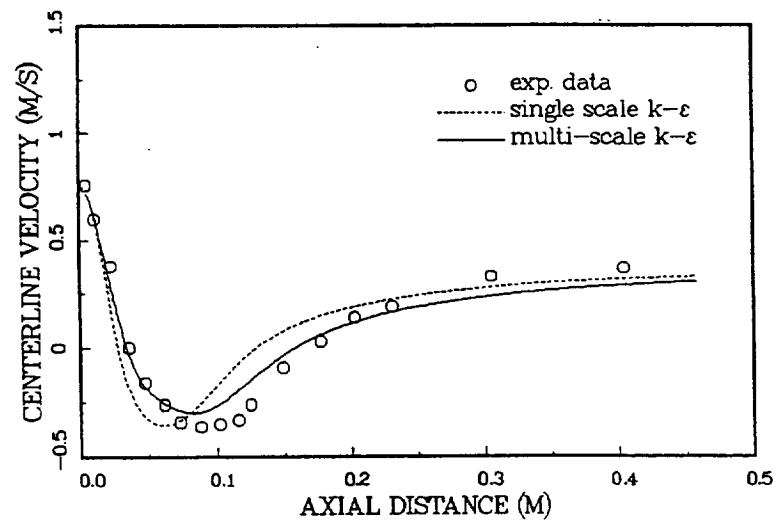


(b) M-S model

Figure 3. Stream-function contours of confined swirling jet flow using the two-layer near wall treatment



(a)



(b)

Figure 4. Axial mean velocity along the centerline
 (a) wall function model
 (b) two-layer model

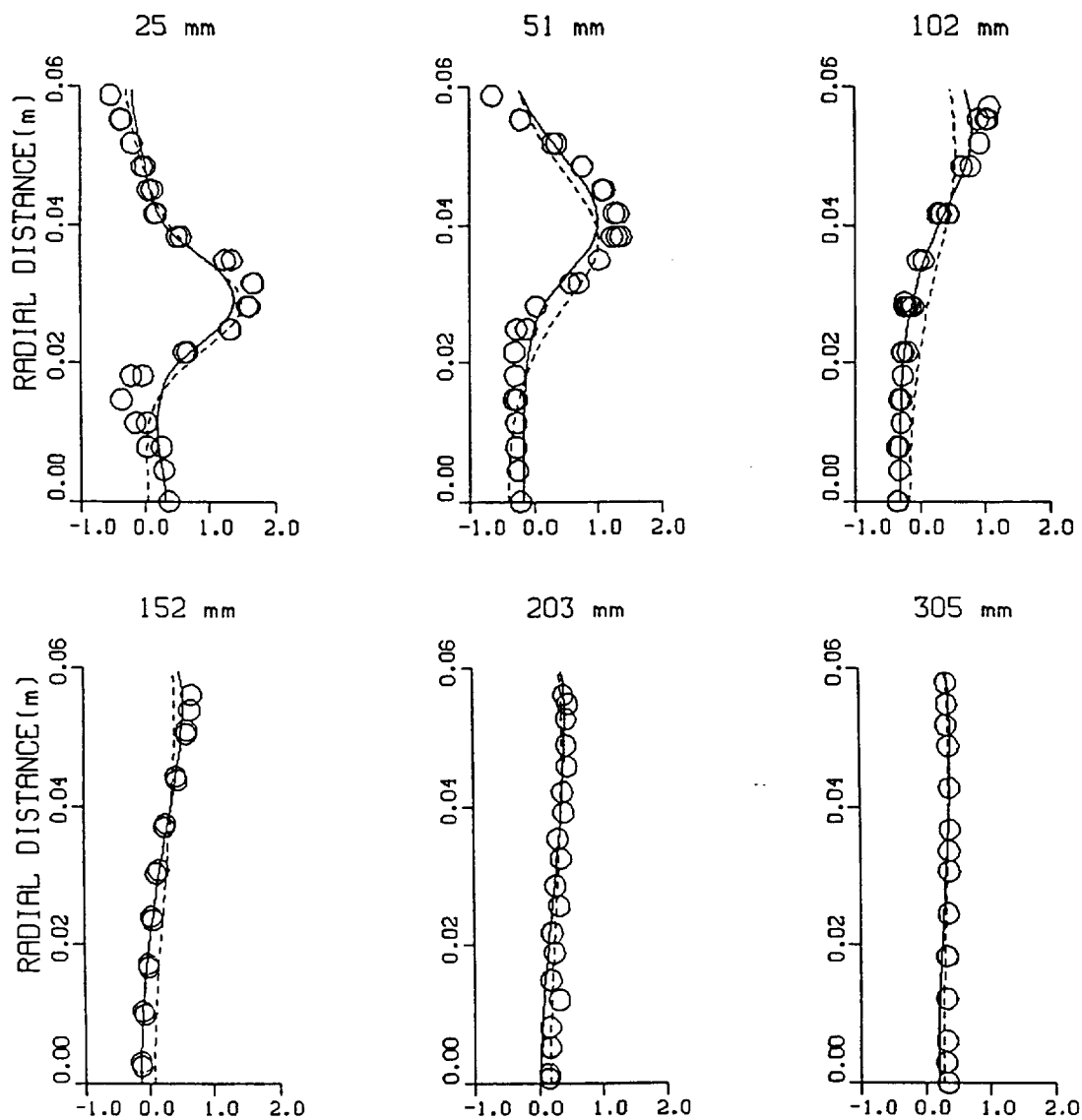


Figure 5. Radial profiles of mean axial velocity using the wall function near wall treatment

- o exp. data
- single scale $k-\epsilon$
- multi-scale $k-\epsilon$

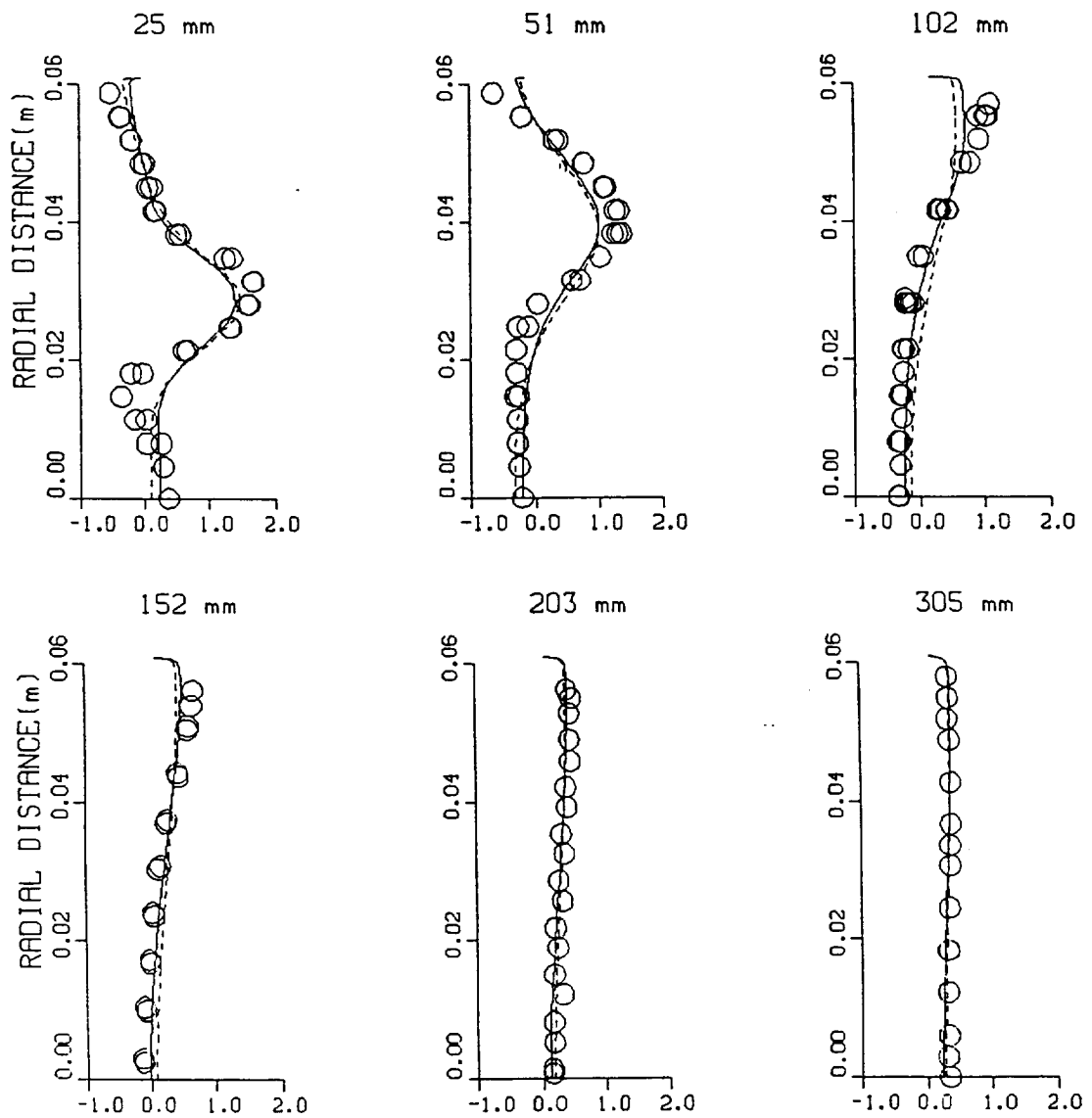


Figure 6. Radial profiles of mean axial velocity using the two-layer near wall treatment

- o exp. data
- single scale $k-\epsilon$
- multi-scale $k-\epsilon$

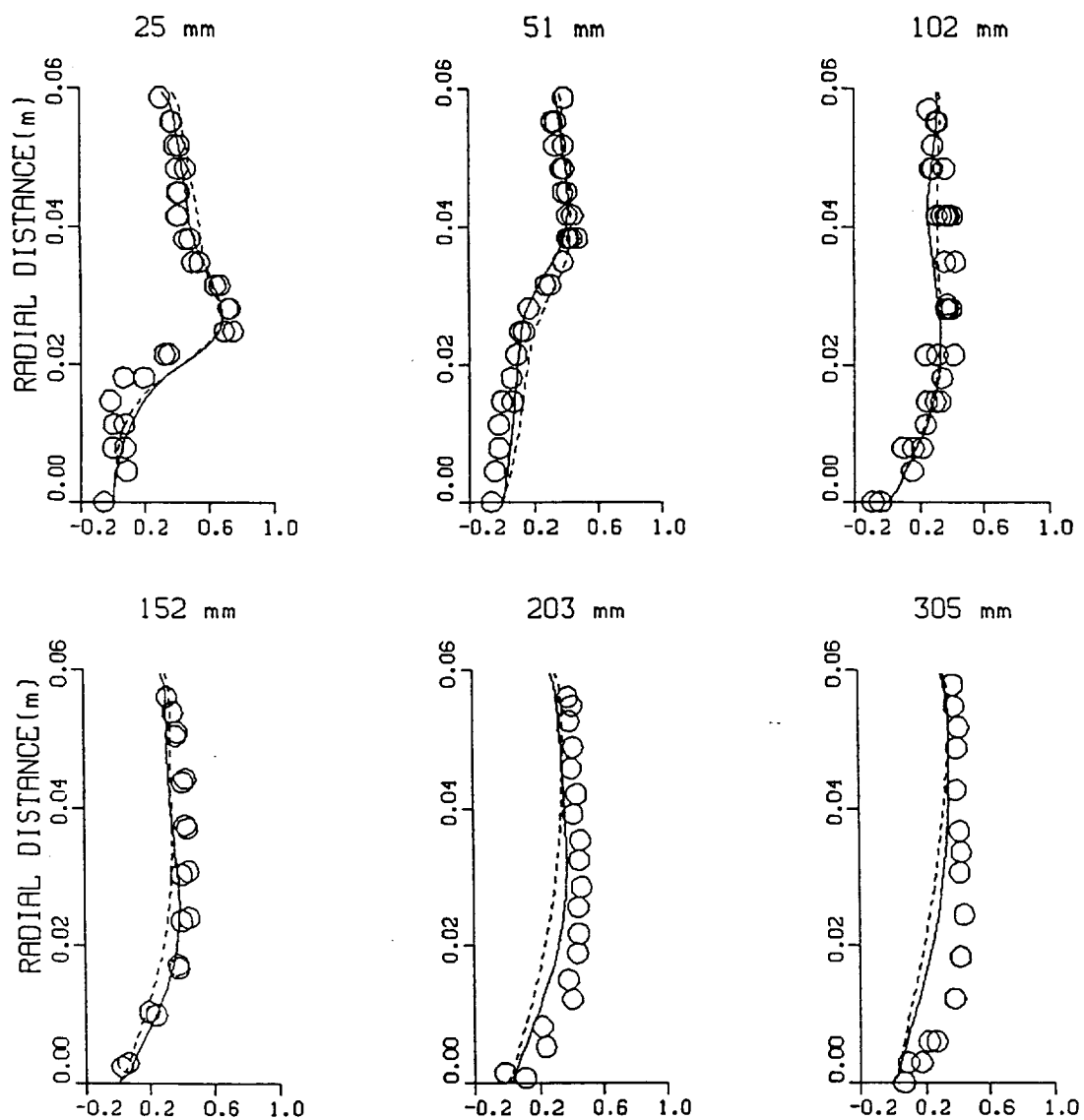


Figure 7 Radial profiles of mean tangential velocity using the wall function near wall treatment

- o exp. data
- single scale $k-\epsilon$
- multi-scale $k-\epsilon$

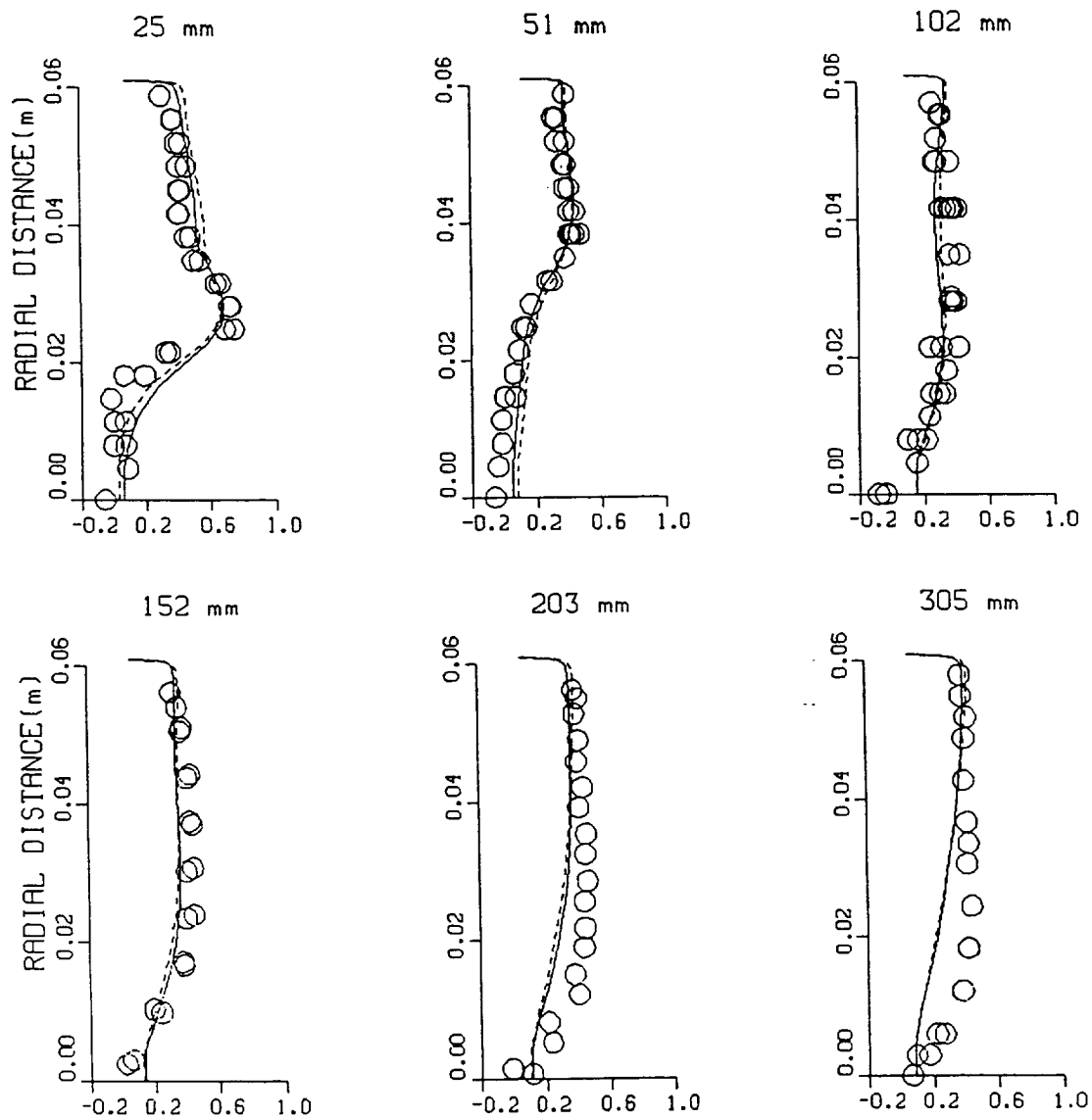


Figure 8. Radial profiles of mean tangential velocity using the two-layer near wall treatment

- o exp. data
- single scale $k-\epsilon$
- multi-scale $k-\epsilon$

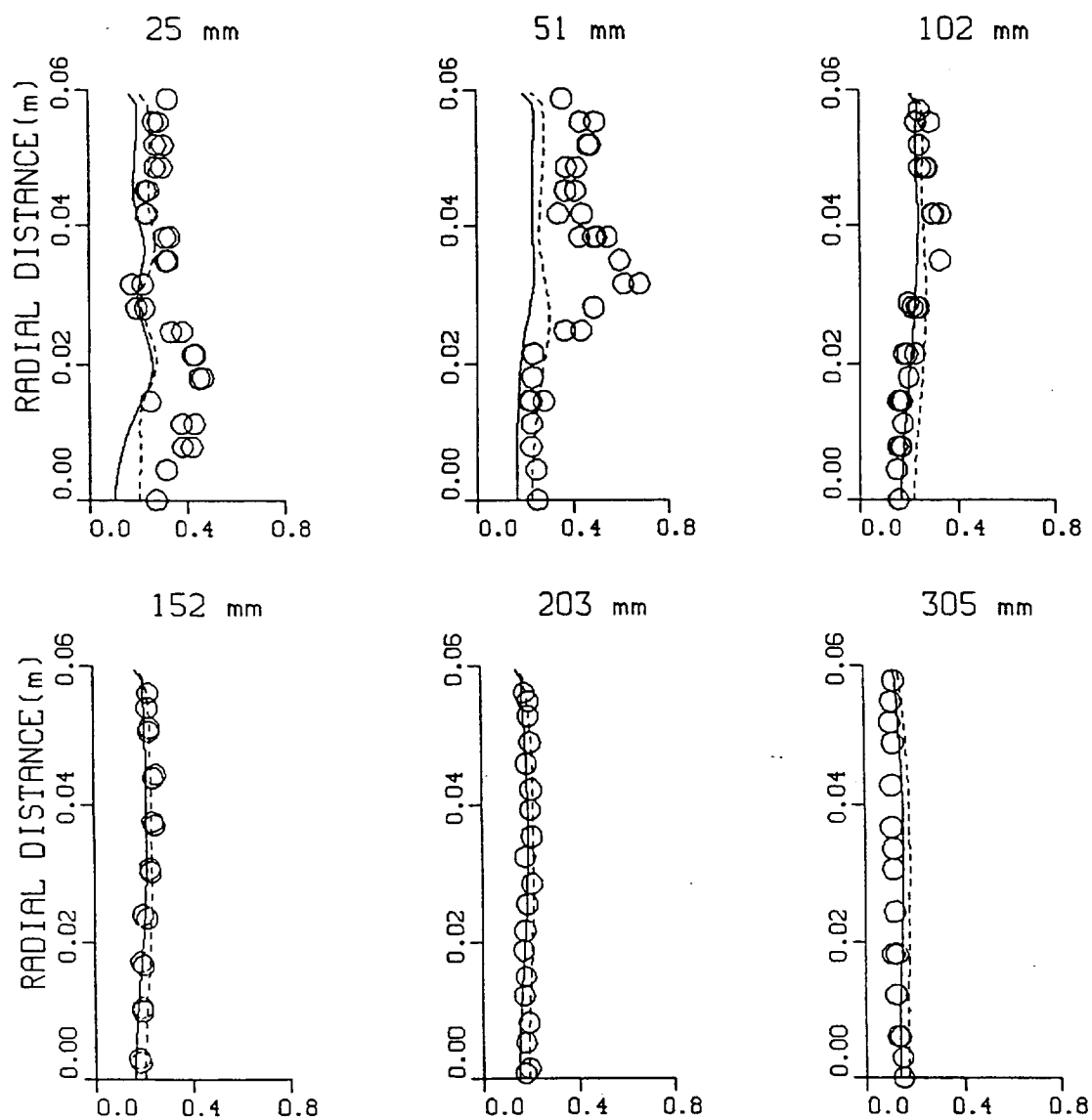


Figure 9. Radial profiles of turbulent intensity $(\overline{uu})^{1/2}$ using the wall-function near wall treatment

- o exp. data
- single scale $k-\epsilon$
- multi-scale $k-\epsilon$

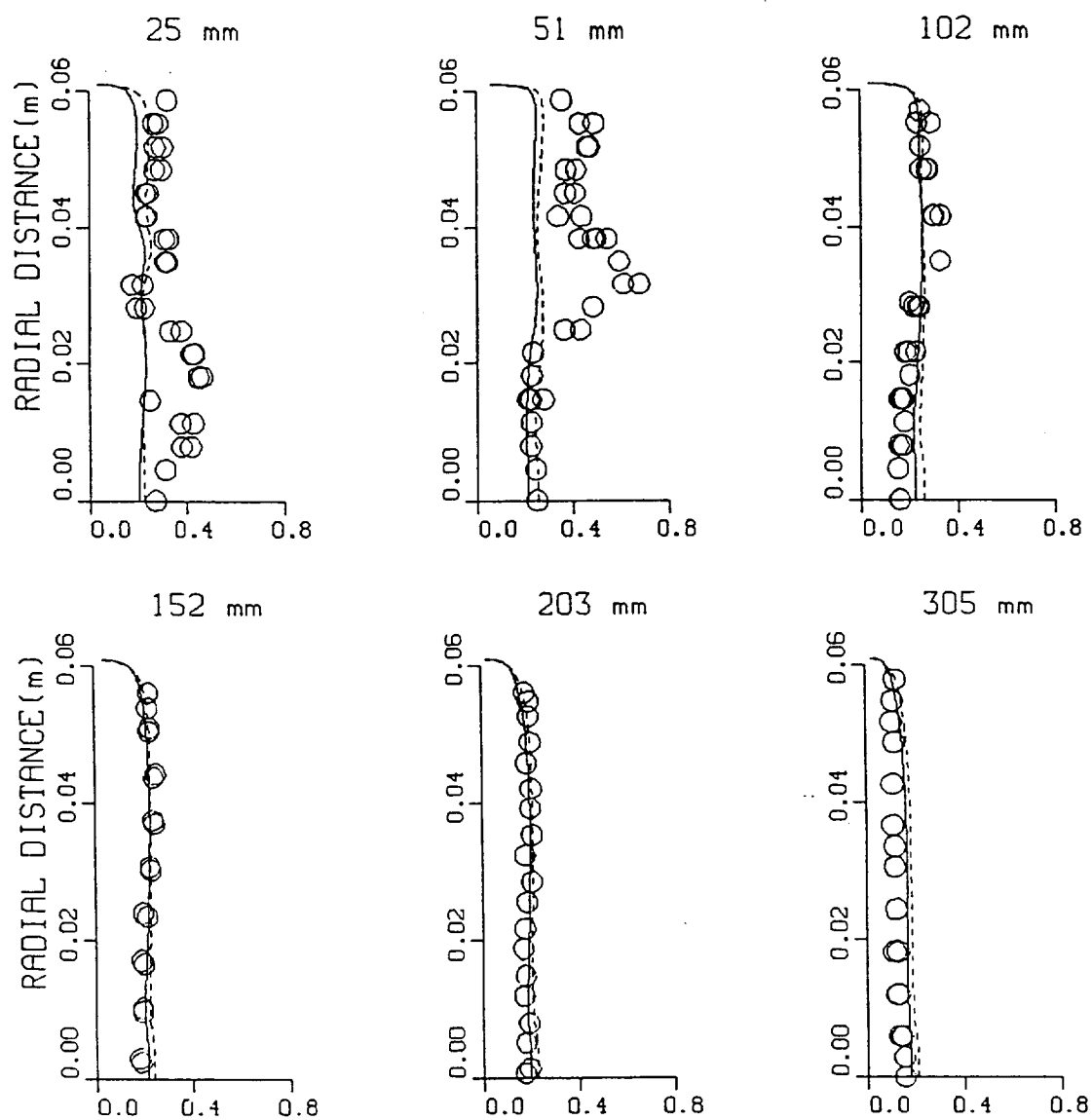


Figure 10. Radial profiles of turbulent intensity

$(\overline{uu})^{1/2}$ using the two layer near-wall treatment

- o exp. data**
- single scale $k-\epsilon$**
- multi-scale $k-\epsilon$**

APPENDIX B

Multi-Scale k - ε Module Deck

B.1 Introduction

This user's manual describes the multi-scale k - ε module deck. The module is a self contained FORTRAN source code to compute turbulent kinetic energy, energy dissipation and turbulent eddy viscosity using the multi-time-scale k - ε turbulence model. It uses as input the mean flow properties as computed by conventional CFD techniques. The module is constructed to be self-contained, stand alone and compatible with a number of CFD solvers. A discussion of the multi-time-scale k - ε module structure is given next together with flow charts to show how to interface the module with a number of flow solvers. A list of variable names used is also given.

B.2 Program KEMOD

This is basically the solver for the k and ε - transport equations in both the production and the dissipation regions of the energy spectrum. It reads through its argument list different variables from the calling flow solver. These variables are described below where, each variable name ends with either an (I) for Integer variable, (R) for Real variable or (L) for Logical variable.

The flow chart of the program is shown in Figure B.1. It shows the main operations performed by the code.

List of Argument Variable Names

XR	Grid node locations in the x or ξ -direction, dimensioned to XR (NX,NY) (input from flow solver)
YR	Grid node locations in the y or η -direction, dimensioned to YR (NX,NY) (input from flow solver)

UR	Axial or x-direction velocity (u), dimensioned as UR (NX,NY) (input from flow solver)
VR	Radial or y-direction velocity (v), also dimensional as VR (NX,NY) (input from flow solver)
WR	Azimuthal velocity (w), dimensional WR (NX,NY) (input from flow solver)
TER	Large scale turbulence kinetic energy k_p , dimensioned TER (NX,NY) (calculated in KEMOD and returned to flow solver)
EDR	Large scale turbulent energy dissipation rate ε_p , dimensioned EDR (NX,NY) (calculated in KEMOD and returned to flow solver)
TETR	Small scale turbulence kinetic energy k_t , dimensioned TETR (NX,NY) (calculated in KEMOD and returned to flow solver)
EDTR	Small scale turbulent energy dissipation rate ε_t , dimensioned EDTR (NX,NY) (calculated in KEMOD and returned to flow solver)
DENR	Fluid density, dimensioned DENR (NX,NY)
URFKER	Under-relaxation factors dimensioned as URFKER(4) and specified as follows: URFKER(1) for large scale turbulent energy equation URFKER(2) for small scale turbulent energy equation URFKER(3) for large scale turbulent energy dissipation equation URFKER(4) for small scale turbulent energy dissipation equation
PRTKER	Prandtl/Schmidt numbers dimensioned as PRTKER(4) and specified as follows: PRTKER(1) for large scale turbulent energy equation PRTKER(2) for small scale turbulent energy equation PRTKER(3) for large scale turbulent energy dissipation equation PRTKER(4) for small scale turbulent energy dissipation equation
GR	= 1.0 if second order upwinding is desired

= 0.0 if first order upwinding is used (input from flow solver).

F1R	Mass flux variable at cell faces in x- or ξ -direction, dimensioned F1R (NX,NY) (input from flow solver)
F2R	Mass flux variable at cell faces in y or η -direction, dimensioned F2R (NX,NY) (input from flow solver)
ITERI	Iteration number (input from flow solver), this number must be equal to 1 for a restart case
VISCOSR	Dynamic viscosity (input from flow solver)
VISR	Eddy viscosity, dimensioned VISR (NX,NY) (calculated in KEMOD and returned to main solver)
URFVISR	Under-relaxation factor for total viscosity calculation
AKSIL	Logical variable for axisymmetric geometry (AKSIL= TRUE) or plain geometry (AKSIL= FALSE) (input from flow solver)
C1R	Turbulence model constant, C_1 (input from flow solver)
C2R	Turbulence model constant, C_2 (input from flow solver)
CMUR	Turbulence model constant, C_μ (input from flow solver)
NIMI	Number of cell nodes in the I- or ξ -coordinate lines. (input from flow solver)
NJMI	Number of cell nodes in the J- or η -coordinate lines. (input from flow solver)
JTBEI	Boundary condition flag along east boundary must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall e.g., for an outlet boundary condition on the east boundary set JTBEI to NJ*2, and similarly for other boundaries, dimensioned JTBEI (NY) (input from flow solver)

JTBWI	Boundary condition flag along west boundary, dimensioned JTBWI (NY) (input from flow solver)
ITBNI	Boundary condition flag along north boundary, dimensioned ITBNI (NX) (input from flow solver)
ITBSI	Boundary condition flag along south boundary, dimensioned ITBSI (NY) (input from flow solver)

Program KEMOD is interfaced with the main flow solver by a call to KEMOD with its arguments. For iterative flow solvers KEMOD is called within the iteration sequence after the solution of the momentum equations where the mean velocities are passed to KEMOD. There are different flow solvers utilizing different schemes from staggered to nonstaggered grid arrangement and for nonorthogonal coordinate system there are at least three alternatives to the choice of the velocity components

- i. Cartesian velocity components
- ii. Contravariant velocity components
- iii. Covariant velocity components

The Cartesian velocity components are the most widely used and have the advantage of simple formulation of the governing equations. Whatever the arrangement used, mass fluxes at cell faces are required and passed to KEMOD as F1R and F2R in both directions. The location of other variables such as k and ϵ are at the cell center or cell nodes.

The module starts by reassigning variable names passed to it from flow solver to names that are shared with the different subroutines of the module in an include file "mske.h". The user must set the values for NX and NY in mske.h greater than or equal to the maximum grid dimensions. Then a check is made if it is the first iteration in which case the grid file "GRIDG" is called -after passing the grid node locations XR & YR in KEMOD- in order to calculate grid related quantities which will be explained later. The need to call GRIDG can be waived if all the grid data are passed to the module. That is all the information about the grid such as interpolation factors FX and FY, cell areas (ARE) and volumes (VOL) and normal distances of first grid point from grid boundaries

(DNS from south boundary, DNN - from north boundary, DNW - from west boundary and DNE - from east boundary).

After this, two calls to subroutine CALCKE are made to calculate the large and small scale turbulent kinetic energies with the identifier IPHI=1 and 2 respectively). The large and small scale energy dissipation equations are solved next by calling subroutine CALCKE again with the identifiers IPHI=3 and 4 respectively. The effective viscosity is calculated next. At locations where ϵ is close to zero (i.e., $\leq 10^{-30}$) viscosity is set to zero. A provision is made for under relaxing changes in effective viscosity which may help to stabilize oscillations and improve convergence rate.

B.3 Subroutines

GRIDG

Before calling this subroutine, the coordinates of all grid nodes, defined in reference to a fixed Cartesian coordinate frame are read. Figure B.2 shows the position of cell and grid nodes.

This subroutine is called only once to calculate coordinates of grid nodes (intersection of grid lines) and geometrical properties of the grid (cell areas and volumes, interpolation factors, normal distances of near-boundary cell nodes from boundary). All variables including grid node coordinates are converted to one-dimensional arrays. These are formed by scanning the grid in J-direction (figure B.2) for I=1, and then repeating for all I's. The position of any node in one-dimensional array is therefore defined as;

$$IJ = (I,J) = (I-1) * NJ + J$$

the actual number of grid nodes is one row and one column less than for all cell nodes. For I = NI and J = NJ fictitious grid nodes are introduced which have the same coordinates as actual nodes on NI-1 in I-direction and NJ-1 in J-direction.

The subroutine then calculates interpolation factors which are associated with cell nodes and are used in the main program to calculate values of dependent variables at locations other than cell nodes (cell centers). Definition of these are given in Figure B.3. Cell areas and volumes are

calculated next followed by calculations of normal distances of near-boundary nodes from all four outer boundaries.

CALCKE (PHI, IPHI)

This subroutine solves the linearized and discretized transport equations for the turbulent energies (k_p and k_t) and the energy dissipation (ϵ_p and ϵ_t). The two dummy parameters in the calling statement, PHI and IPHI, represent arrays containing dependent variables for which the equation is to be solved. the subroutine sets up the convective and diffusive coefficients over the entire field. Then it calculates the source terms for either k or ϵ transport equations. A call is made to entry MODMSKE in order to modify these sources and boundary coefficients to suit the particular problem.

The discretized equations have the form

$$A_p \Phi_p = \sum_{i=E,W,N,S} A_i \Phi_i + S_\Phi$$

where the coefficients A_i ($i=E,W,N,S$ see figure B.3) contain both the convective and diffusive fluxes. these equations are assembled and solved by calling subroutine SOLSIP which is based on Stone's Strongly Implicit Solver [7].

SOLSIP

This subroutine solves the system of linear algebraic equations for k and ϵ using Stone's Implicit Procedure [7]. The array RES (IJ) is used to store residuals. The sum of absolute residuals "RESORP" calculated in the first pass through this part of the routine is used as a measure of convergence of the solution process as a whole and this value is stored in RESOR (IPHI). This variable RESOR (IPHI) is passed to the main solver and if desired can be normalized and compared with the maximum error allowed there. If necessary inner iterations counter L and the sum of absolute residuals RESORP are printed out to monitor the rate of convergence of k and ϵ solution. If the ratio RSM is greater than the maximum allowed for the variable in question, SOR (IPHI), and the number of inner iterations is smaller than a prescribed maximum, NSWP (IPHI),

then the routine repeats the sequence of calculating the residuals, increment vectors and updating the dependent variable.

MODMSKE

This subroutine is called from CALCKE subroutine and sets the boundary conditions for k_p , k_t and ε_p , ε_t depending on which variable being called (IDIR = 1, 2, 3, and 4 for k_p , k_t , ε_p , and ε_t respectively). Consider the south boundary for example, if it is one of four options:

- (1) An inflow boundary ITBS(I) = 1, where the source term is set to accept the inlet values at J = 1 (south boundary)
- (2) Outflow boundary ITBS(I) = 2, where zero gradient in y or η -direction is employed.
- (3) Symmetry boundary, TBS(I) = 3, where gradients normal to symmetry plane are zero.
- (4) Wall boundary, ITBS(I) = 4, where the production term GENTS(I) calculated from subroutine WALLFN in program MODIFY is added to the rest of the source term SU(IJ).

B.4 Program MODIFY

This subroutine is called from the u and v solver routines. It basically updates the flux source term of the discretized momentum equation due to wall shear stresses. If the u-momentum equation for example is discretized in the form

$$A_p^* u_p = \sum_{i=EWSN} A_i u_i + S_u^*$$

where P, E, W, N, S are cell nodes as shown in Figure B.3, and A_p^* and A_i 's contain convective and diffusive coefficients. S_u^* is the source term containing pressure gradients and cross-derivative diffusion terms and convective terms for second-order upwinding scheme. This source term is

usually linearized as $S_u^* = S_u - B_p u_p$. The term B_p is usually moved to the left hand side of the equation and modifies the diagonal coefficient $A_p = A_p^* + B_p$, and the equation can be written as

$$A_p u_p = \sum_{i=EWNS} A_i u_i + S_u$$

Then S_u and B_p are passed to subroutine MODIFY where they are modified if a wall is present (e.g., ITBS(I) = 4 for south boundary).

List of Argument Variable Names

CMU	Turbulence model constant, C_μ (input from flow solver)
VISCOS	Dynamic viscosity (input from flow solver)
XX	Grid node locations in the x or ξ -direction, dimensioned to XX (NX*NY) (input from flow solver)
YY	Grid node locations in the y or η -direction, dimensioned to YY (NX*NY) (input from flow solver)
R	Grid node radius equal to 1 for non-axisymmetric and YY for axisymmetric, dimensioned to R (NX*NY) (input from flow solver)
DNS	Normal distance to south, dimensioned to DNS (NX*NY) (input from flow solver)
DNN	Normal distance to north, dimensioned to DNN (NX*NY) (input from flow solver)
DNW	Normal distance to west, dimensioned to DNW (NX*NY) (input from flow solver)
DNE	Normal distance to east, dimensioned to DNE (NX*NY) (input from flow solver)

U	Axial or x-direction velocity (u), dimensioned as UR (NX*NY) (input from flow solver)
V	Radial or y-direction velocity (v), also dimensional as VR (NX*NY) (input from flow solver)
W	Azimuthal velocity (w), dimensional WR (NX*NY) (input from flow solver)
DEN	Fluid density, dimensional DEN (NX*NY) (input from flow solver)
TE	Large scale turbulence kinetic energy k_p , dimensioned TE (NX*NY) (calculated in KEMOD and returned to flow solver)
TET	Small scale turbulence kinetic energy k_t , dimensioned TET (NX*NY) (calculated in KEMOD and returned to flow solver)
SU	Variable source term, dimensioned SU (NX*NY)
BP	Constant source term, dimensioned BP (NX*NY)
AE	Cell area, dimensioned to AE (NX*NY) (input from flow solver)
AW	Cell area, dimensioned to AW (NX*NY) (input from flow solver)
AN	Cell area, dimensioned to AN (NX*NY) (input from flow solver)
AS	Cell area, dimensioned to AS (NX*NY) (input from flow solver)

SUVS,SPVS,SUWS,SPWS

Source terms at south boundary due to wall shear stress, all dimensioned to S##S (NX*NY) (returned to flow solver)

SUVN,SPVN,SUWN,SPWN

Source terms at north boundary due to wall shear stress, all dimensioned to S##N (NX*NY) (returned to flow solver)

SUVW,SPVW,SUWW,SPWW

Source terms at west boundary due to wall shear stress, all dimensioned to S##W (NX*NY) (returned to flow solver)

SUVE,SPVE,SUWE,SPWE

Source terms at east boundary due to wall shear stress, all dimensioned to S##E (NX*NY) (returned to flow solver)

GENTS,GENTN,GENTW,GENTEE

Generation terms at south, north, west, and east boundaries respectively due to moving walls, with GENTS(NX), GENTN(NX), GENTW(NY), and GENTEE(NY) (returned to flow solver)

NX Maximum number of cell nodes in the I- or ξ -coordinate lines. (input from flow solver)

NY Maximum number of cell nodes in the J- or η -coordinate lines. (input from flow solver)

NXNY NX*NY

NIM Number of cell nodes in the I- or ξ -coordinate lines. (input from flow solver)

NJM Number of cell nodes in the J- or η -coordinate lines. (input from flow solver)

ITBS Boundary condition flag along south boundary must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall e.g., for an outlet boundary condition on the east boundary set ITBS to NI*2, and similarly for other boundaries, dimensioned ITBS (NX) (input from flow solver)

ITBN Boundary condition flag along north boundary, dimensioned ITBN (NX) (input from flow solver)

JTBW Boundary condition flag along west boundary, dimensioned ITBNI (NY) (input from flow solver)

JTBE

Boundary condition flag along east boundary, dimensioned JTBE (NY) (input from flow solver)

For an iterative flow solver using the finite-volume methodology. A typical interface and call to KEMOD from the main flow solver can be represented by a flow chart as shown in figure B.4.

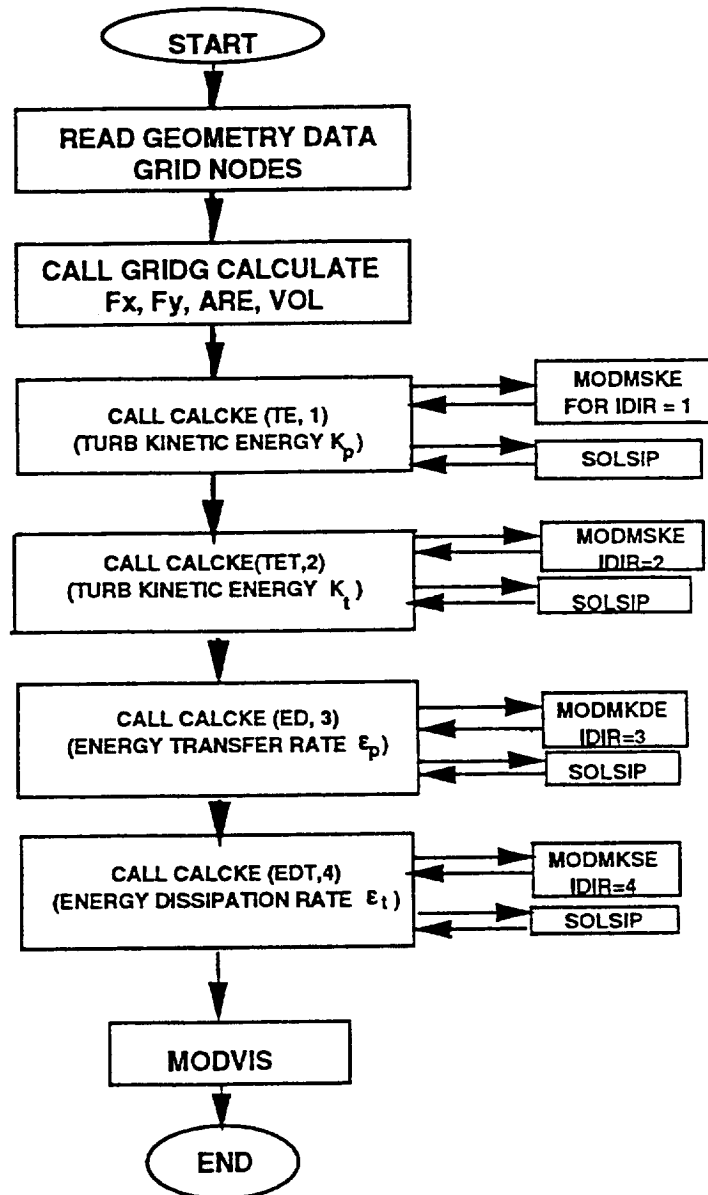


Figure B.1 Multi-scale k - ϵ module deck flow chart

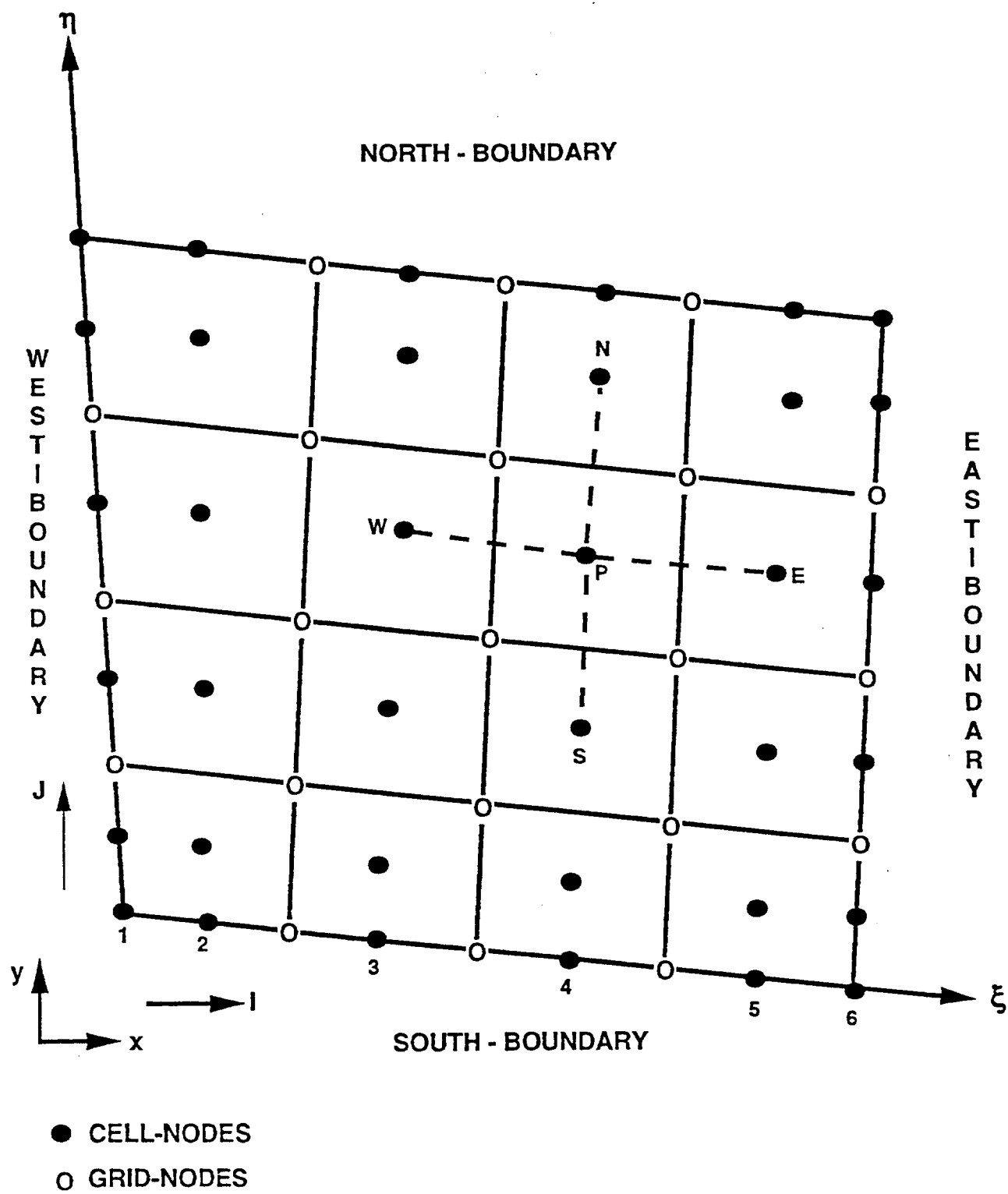
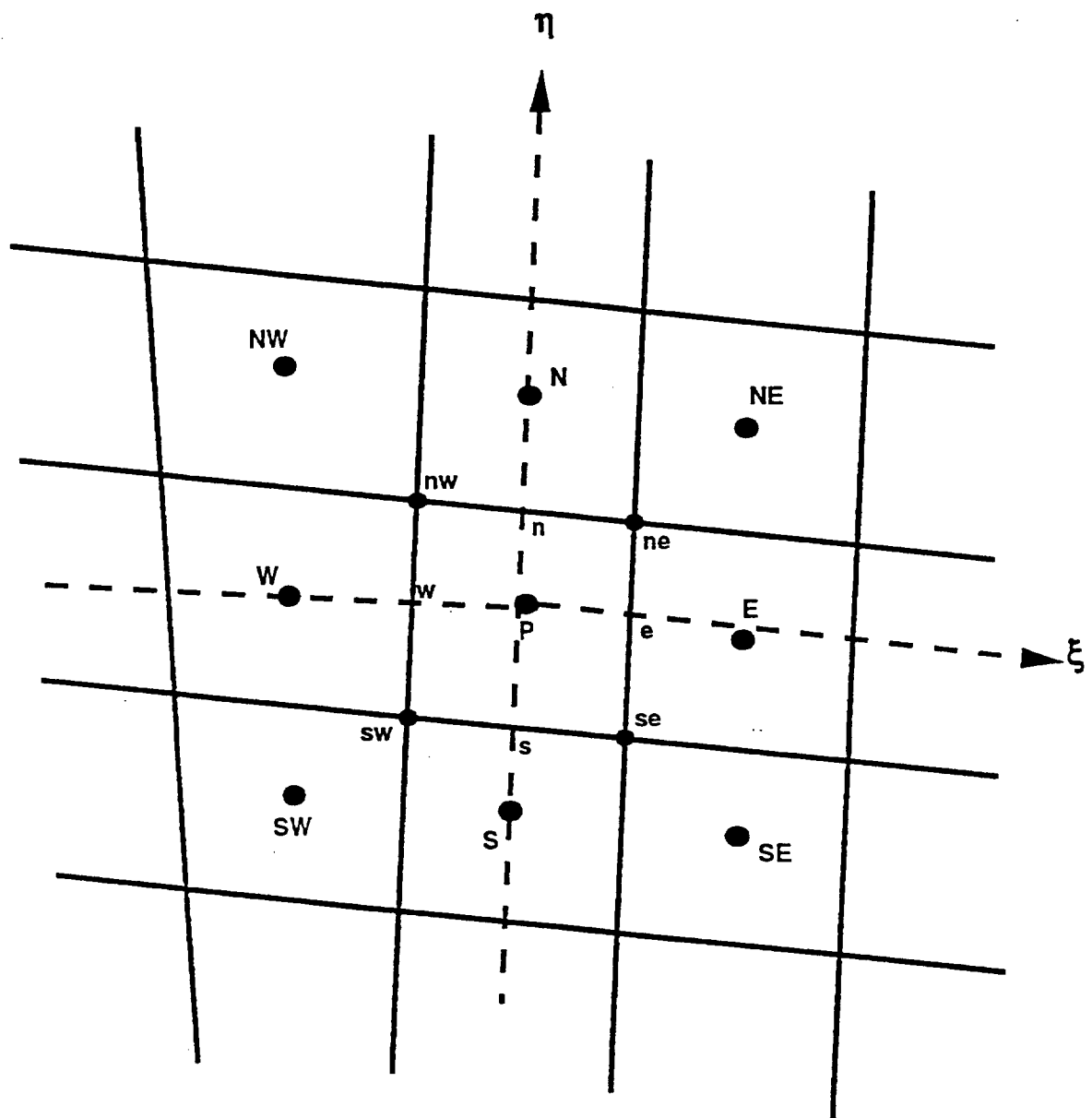


Figure B.2 Position of cell and grid nodes



$$FX_p = \frac{\overline{Pe}}{\overline{Pe} + \overline{eE}}, \quad FY_p = \frac{\overline{Pn}}{\overline{Pn} + \overline{nN}}$$

Figure B.3 Definition of interpolation factors

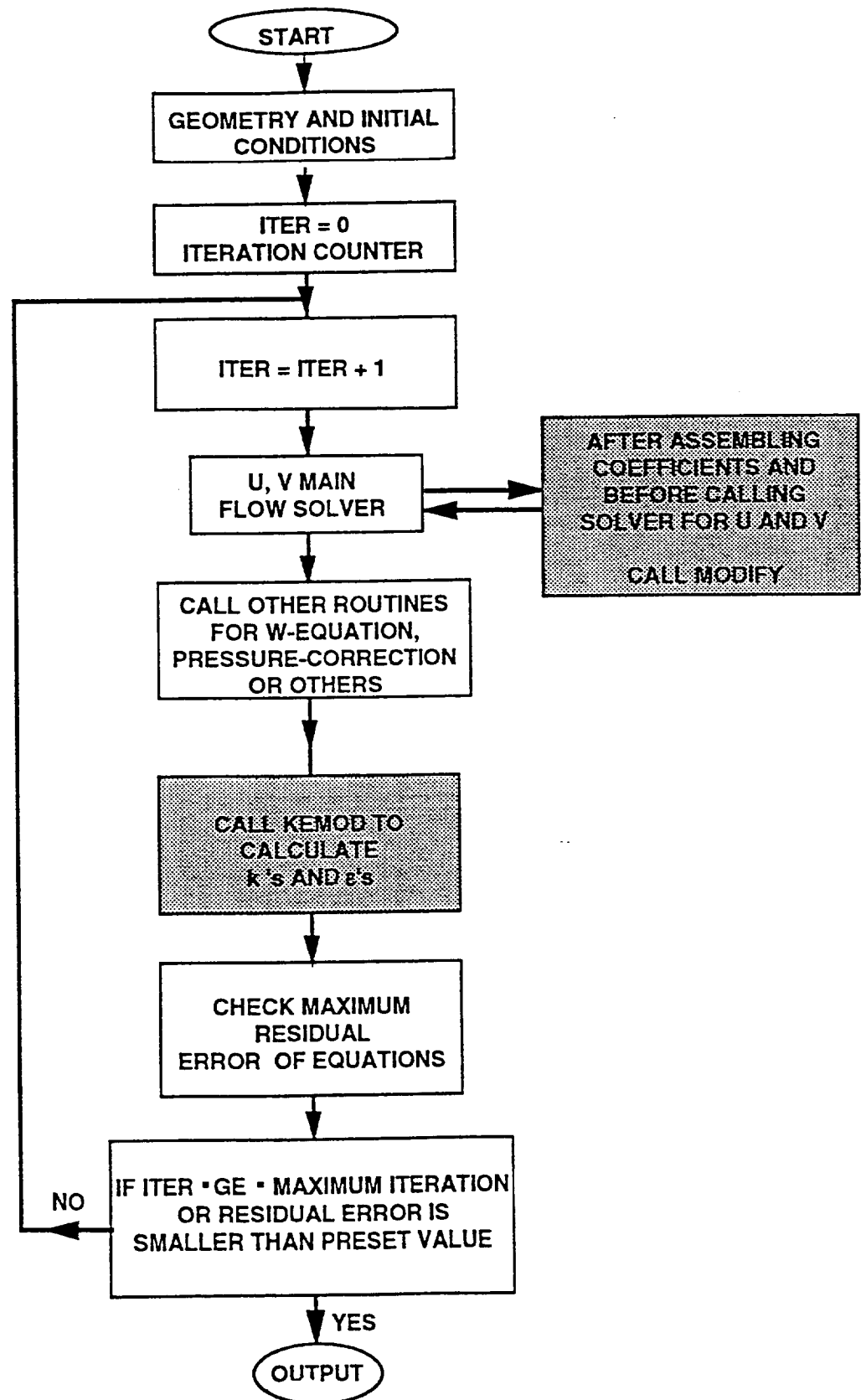


Figure B.4 Typical main flow solver flow chart with calls to the multi-scale k-e module

Oct 12 1996 16:38 mskemod_2d Page 3

```

143 C
144 NJ=NJM+1
145 NI=NIM+1
146 NING=NIM*NJ
147 DO 2 I=1,NI
148   IMNJ(I)=(I-1)*NJ
149   CONTINUE
150 DO 3 I=1,NIM
151   J=NJM
152   X(I,J+1)=X(I,J)
153   Y(I,J+1)=Y(I,J)
154   CONTINUE
155 DO 4 J=1,NJ
156   I=NIM
157   X(I+1,J)=X(I,J)
158   Y(I+1,J)=Y(I,J)
159   CONTINUE
160 C
161 C... GRID ORIGIN AT X=0, Y=0
162 DO 5 I=1,NI
163   DO 5 J=1,NJ
164     IJ=(I-1)*NJ+J
165     XX(IJ)=X(I,J)
166     YY(IJ)=Y(I,J)
167   CONTINUE
168 C
169 C-----CALCULATION OF INTERPOLATION FACTORS
170 C
171 DO 6 IJ=1,NINJ
172   FX(IJ)=0.
173   FY(IJ)=0.
174   CONTINUE
175 C
176 DO 7 J=2,NJM
177   IJ=J
178   FX(IJ)=0.
179   LYB=NIM-1
180   DO 8 I=2,LIE
181     IJ=IMNJ(I)+J
182     IPJ=IJ+NJ
183     IJM=IJ-1
184     IMJ=IJ-NJ
185     DXPE=0.5*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))
186     DYPE=0.5*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
187     DYP=0.5*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
188     DYP=0.5*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))
189     DKP=SQRT(DXPE**2+DYPE**2)
190     DKE=SQRT(DXPE**2+DYPE**2)
191     FX(IJ)=DKP/(DKP+DKE)
192     CONTINUE
193   IJ=IMNJ(NIM)+J
194   FX(IJ)=1.0
195   CONTINUE
196 C
197 DO 9 I=1,NI
198   IJ=IMNJ(I)+1
199   FX(IJ)=FX(IJ+1)
200   IJ=IMNJ(I)+NJ
201   FX(IJ)=FX(IJ-1)
202   CONTINUE
203 C
204 DO 10 I=2,NIM
205   IJ=IMNJ(I)+1
206   FY(IJ)=0.0
207   LJM=NJM-1
208   DO 11 J=2,LJM
209     IJ=IMNJ(I)+J
210     IJP=IJ+1
211     IJM=IJ-1
212     DXP=0.5*(XX(IJ)+XX(IMJ)-XX(IJM)-XX(IMJ-1))
213

```

mskemod_2d

Oct 12 1996 16:38 mskemod_2d Page 4

```

214   DXPN=0.5*(XX(IJP)+XX(IMJ+1)-XX(IJ)-XX(IMJ))
215   DYP=0.5*(YY(IJ)+YY(IMJ)-YY(IJM)-YY(IMJ-1))
216   DYEN=0.5*(YY(IJP)+YY(IMJ+1)-YY(IJ)-YY(IMJ))
217   DEF=SQRT(DXPN**2+DYP**2)
218   DEPN=SQRT(DXPN**2+DYP**2)
219   FY(IJ)=DEF/(DEPN+DEPN)
220   CONTINUE
221   IJ=IMNJ(I)+NJM
222   FY(IJ)=1.0
223   CONTINUE
224 C
225 DO 12 J=1,NJ
226   FY(IJ)=FY(J+NJ)
227   IJ=IMNJ(NI)+J
228   FY(IJ)=FY(IJ-NJ)
229   CONTINUE
230 C
231 C-----CALCULATION OF CELL AREAS
232 C
233 DO 13 IJ=1,NINJ
234   ARE(IJ)=0.
235   CONTINUE
236 C
237 DO 14 I=2,NIM
238   DO 14 J=2,NJM
239     IJ=IMNJ(I)+J
240     DXNWSW=XX(IJ)-XX(IJ-NJ-1)
241     DYNSEW=YY(IJ)-YY(IJ-NJ-1)
242     DXNWSE=XX(IJ-NJ)-XX(IJ-1)
243     DYNWSE=YY(IJ-NJ)-YY(IJ-1)
244     ARE(IJ)=0.5*ABS(DXNWSW*DYNWSE-DXNWSE*DYNESW)
245   CONTINUE
246 C
247 C-----NORMAL DISTANCE FROM THE WALL
248 C
249 DO 15 I=1,NI
250   DNS(I)=0.
251   DNN(I)=0.
252   CONTINUE
253 C
254 DO 16 J=1,NJ
255   DNW(J)=0.
256   DNE(J)=0.
257   CONTINUE
258 C
259 DO 17 I=2,NIM
260   IJ=IMNJ(I)+2
261   IMJ=IJ-NJ
262   DXB=XX(IJ-1)-XX(IMJ-1)
263   DYB=YY(IJ-1)-YY(IMJ-1)
264   DXBP=0.25*(XX(IJ)-XX(IJ-1)+XX(IMJ)-XX(IMJ-1))
265   DYBP=0.25*(YY(IJ)-YY(IJ-1)+YY(IMJ)-YY(IMJ-1))
266   DNS(I)=DELTA(DXB,DYB,DXBP,DYBP)
267   IJ=IMNJ(I)+NJM
268   IMJ=IJ-NJ
269   DXB=XX(IJ)-XX(IMJ)
270   DYB=YY(IJ)-YY(IMJ)
271   DXBP=0.25*(XX(IJ-1)-XX(IMJ-1)-XX(IMJ)-XX(IMJ))
272   DYBP=0.25*(YY(IJ-1)-YY(IMJ-1)-YY(IMJ)-YY(IMJ))
273   DNN(I)=DELTA(DXB,DYB,DXBP,DYBP)
274   CONTINUE
275 C
276 DO 18 J=2,NJM
277   IJ=IMNJ(2)+J
278   IMJ=IJ-NJ
279   DXB=XX(IMJ)-XX(IMJ-1)
280   DYB=YY(IMJ)-YY(IMJ-1)
281   DXBP=0.25*(XX(IJ)-XX(IMJ)+XX(IJ-1)-XX(IMJ-1))
282   DYBP=0.25*(YY(IJ)-YY(IMJ)+YY(IJ-1)-YY(IMJ-1))
283   DNN(I)=DELTA(DXB,DYB,DXBP,DYBP)
284   IJ=IMNJ(NIM)+J

```

Oct 12 1996 16:38	mskmod_2d	Page 5
285	IMJ=IJ-NJ	
286	DXB=XX(IJ)-XX(IJ-1)	
287	DYB=YY(IJ)-YY(IJ-1)	
288	DXBP=0.25*(XX(IMJ)-XX(IJ)+XX(IMJ-1)-XX(IJ-1))	
289	DYBP=0.25*(YY(IMJ)-YY(IJ)+YY(IMJ-1)-YY(IJ-1))	
290	DNE(J)=DELTA(DXB,DYB,DXBP,DYBP)	
291	18 CONTINUE	
292	C	
293	C----- CALCULATE CELL VOLUMES	
294	C	
295	DO 19 IJ=1,NINJ	
296	VOL(IJ)=ARE(IJ)	
297	19 CONTINUE	
298	C	
299	IF(AKSI) THEN	
300	SIXR=1./6.	
301	DO 20 I=2,NIM	
302	I1=IMNJ(I)	
303	DO 21 J=2,NJM	
304	IJ=I1+J	
305	IMJ=IJ-NJ	
306	IMJM=IMJ-1	
307	IJM=IJ-1	
308	RIJ=YY(IJ)**2	
309	RIMJ=YY(IMJ)**2	
310	RIMJM=YY(IMJM)**2	
311	RIJM=YY(IJM)**2	
312	VOL(IJ)=SIXR*((XX(IJ)-XX(IMJ))* (RIJ+RIMJ+YY(IJ)*YY(IMJ))+	
313	& ((XX(IMJ)-XX(IMJM))* (RIMJ+RIMJM+YY(IMJ)*YY(IMJM))+	
314	& ((XX(IMJM)-XX(IJM))* (RIMJM+RIJM+YY(IMJM)*YY(IJM))+	
315	& ((XX(IJM)-XX(IJ))* (RIJM+RIJ+YY(IJM)*YY(IJ)))	
316	21 CONTINUE	
317	20 CONTINUE	
318	ENDIF	
319	C	
320	C----- INITIALIZE VARIABLES INITIALLY	
321	C	
322	HAF=0.5	
323	QPR=0.25	
324	SMAL=1.E-30	
325	GRAT=1.E30	
326	DO 22 IJ=1,NINJ	
327	DEN(IJ)=DENSIT	
328	VIS(IJ)=VISCOS	
329	FMU(IJ)=1.0	
330	FLR1(IJ)=1.0	
331	FLR2(IJ)=1.0	
332	APU(IJ)=0.0	
333	APV(IJ)=0.0	
334	AE(IJ)=0.0	
335	AS(IJ)=0.0	
336	AN(IJ)=0.0	
337	AW(IJ)=0.0	
338	BE(IJ)=0.0	
339	BW(IJ)=0.0	
340	BN(IJ)=0.0	
341	BS(IJ)=0.0	
342	RES(IJ)=0.0	
343	R(IJ)=1.0	
344	22 CONTINUE	
345	IF(AKSI) THEN	
346	DO 23 IJ=1,NINJ	
347	R(IJ)=YY(IJ)	
348	23 CONTINUE	
349	ENDIF	
350	C	
351	RETURN	
352	END	
353	C	
354	C	
355	C*****	

Oct 12 1996 16:38	mskmod_2d	Page 6
356	SUBROUTINE CALCE (PHI,IPHI)	
357	C*****	
358	C	
359	INCLUDE 'mskmod.h'	
360	C	
361	DIMENSION UW(NY),VW(NY),WW(NY),PHI(NXNY),FXW(NY),DW(NY)	
362	C	
363	GO TO (1,2,3,4) IPHI	
364	CONTINUE	
365	URPHI=UREFKP	
366	PRTNVP=1./PRTKP	
367	GO TO 5	
368	CONTINUE	
369	URPHI=UREFKT	
370	PRTNVP=1./PRTKT	
371	GO TO 5	
372	CONTINUE	
373	URPHI=URFED	
374	PRTNVP=1./PRTED	
375	GO TO 5	
376	CONTINUE	
377	URPHI=URFEDT	
378	PRTNVP=1./PRTEDT	
379	5 CONTINUE	
380	C	
381	IJ=1	
382	PHINE=PHI(IJ)	
383	PHINW(IJ)=PHINE	
384	C	
385	DO 10 J=2,NJM	
386	IJ=J	
387	IJM=IJ-1	
388	IPJ=IJ+NJ	
389	IJP=IJ+1	
390	FYN=FY(IJ)	
391	FYS=1.0-FYN	
392	AREE=HAF*(ARE(IJ)+ARE(IPJ))	
393	DYE=XX(IJ)-XX(IJM)	
394	DYE=YY(IJ)-YY(IJM)	
395	VIST=VIS(IJ)-VISCOS	
396	GAME=HAF*(VISCOS+VIST*PRTNV(IPHI))*(R(IJ)+R(IJM))	
397	DW(IJ)=GAME/AREE*(DXE**2+DYE**2)	
398	PHISE=PHINE	
399	PHINE=PHI(IJ+1)*FYN+PHI(IJ)*FYS	
400	PHINW(IJ)=PHINE	
401	UW(IJ)=U(IJ)	
402	VW(IJ)=V(IJ)	
403	WW(IJ)=W(IJ)	
404	SNSW(IJ)=0.	
405	FXW(IJ)=1.0	
406	IF(JTBW(IJ).EQ.3.OR.JTBW(IJ).EQ.4) GO TO 10	
407	DXKS=QPR*(XX(IPJ)+XX(IPJ-1)-XX(IJ)-XX(IJM))	
408	DYKS=QPR*(YY(IPJ)+YY(IPJ-1)-YY(IJ)-YY(IJM))	
409	SNSW(IJ)=-GAME/AREE*(DXKS+DYE+DYKS+DYE)*(PHINE-PHISE)	
410	10 CONTINUE	
411	C	
412	DO 100 I=2,NIM	
413	J=1	
414	IJ=IMNJ(I)+J	
415	IMJ=IJ-NJ	
416	IJP=IJ+1	
417	FXE=FX(IJ)	
418	FXW=1.-FX(IJ)	
419	AREN=HAF*(ARE(IJ)+ARE(IJP))	
420	DXN=XX(IJ)-XX(IMJ)	
421	DYN=YY(IJ)-YY(IMJ)	
422	VIST=VIS(IJ)-VISCOS	
423	GAMN=HAF*(VISCOS+VIST*PRTNV(IPHI))*(R(IJ)+R(IMJ))	
424	DN=GAMN/AREN*(DXN**2+DYN**2)	
425	FYSS=1.0	
426	PHINE=PHI(IJ+NJ)*FXE+PHI(IJ)*FXW	

Oct 12 1996 16:38	mskmod_2d	Page 7
427	UN=U(IJ)	
428	VN=V(IJ)	
429	WN=W(IJ)	
430	SEWN=0.	
431	IF (ITBS(I).EQ.3.OR.ITBS(I).EQ.4) GO TO 110	
432	DXET=QTR*(XX(IJP)+XX(IJP-NJ)-XX(IJ)-XX(IMJ))	
433	DYET=QTR*(YY(IJP)+YY(IJP-NJ)-YY(IJ)-YY(IMJ))	
434	SEWN=GAMN/AREN*(DXN*DXET+DYN*DYET)*(PHINE-PHINW(J))	
435	SEWNBS=SEWN	
436	110 CONTINUE	
437	PHINW(J)=PHINE	
438	C	
439	C-----THE MAIN LOOP - ASSEMBLY OF COEFFICIENTS AND SOURCES	
440	C	
441	DO 101 J=2,NJM	
442	IJ=IMNJ(I)+J	
443	IPJ=IJ+NJ	
444	IMJ=IJ-NJ	
445	IJP=IJ+1	
446	IJM=IJ-1	
447	FXE=FX(IJ)	
448	FXW=1.-FXE	
449	FYN=FY(IJ)	
450	FYS=1.-FYN	
451	DYE=XX(IJ)-XX(IJM)	
452	DYE=YY(IJ)-YY(IJM)	
453	DYN=XX(IJ)-XX(IMJ)	
454	DYN=YY(IJ)-YY(IMJ)	
455	AREE=HAF*(ARE(IJ)+ARE(IPJ))	
456	AREN=HAF*(ARE(IJ)+ARE(IJP))	
457	WISE=VIS(IJ)*FXW+VIS(IPJ)*FXE	
458	WISE=VIS(IJ)*FYN+VIS(IPJ)*FYS	
459	GAMN=HAF*(VISCOS+WISE*PRTNV(IPHI))*(R(IJ)+R(IJM))	
460	VISN=VIS(IJ)*FYS+VIS(IJP)*FYN	
461	VISNT=VISN-VISCO	
462	GAMN=HAF*(VISCOS+VISNT*PRTNV(IPHI))*(R(IJ)+R(IMJ))	
463	C	
464	DS=DN	
465	DE=GAME/AREE*(DXE**2+DYE**2)	
466	DN=GAMN/AREN*(DXN**2+DYN**2)	
467	C	
468	C LINEAR UPWIND DIFFERENCING	
469	C	
470	AEP=MIN(F1(IJ),0.0)*FX(IPJ)*G	
471	AWW=-MAX(F1(IMJ),0.0)*(1.0-FXW(J))*G	
472	AE1=-MIN(F1(IMJ),0.0)*FXE*G	
473	AW1=MAX(F1(IJ),0.0)*(1.0-FX(IMJ))*G	
474	ANN=MIN(F2(IJ),0.0)*FY(IJP)*G	
475	ASS=-MAX(F2(IJM),0.0)*(1.0-FY(J))*G	
476	AN1=-MIN(F2(IJM),0.0)*FYN*G	
477	AS1=MAX(F2(IJ),0.0)*(1.0-FY(IJM))*G	
478	C	
479	AW(IJ)=DW(J)+MAX(F1(IMJ),0.0)-AWW	
480	AE(IJ)=DE-MIN(F1(IJ),0.0)-AREE	
481	AS(IJ)=DS-MAX(F2(IJM),0.0)-ASS	
482	AN(IJ)=DN-MIN(F2(IJ),0.0)-ANN	
483	C	
484	DXKS=QTR*(XX(IPJ)-XX(IMJ)+XX(IPJ-1)-XX(IMJ-1))	
485	DYKS=QTR*(YY(IPJ)-YY(IMJ)+YY(IPJ-1)-YY(IMJ-1))	
486	DXET=QTR*(XX(IJP)-XX(IJM)+XX(IJP-NJ)-XX(IJM-NJ))	
487	DYET=QTR*(YY(IJP)-YY(IJM)+YY(IJP-NJ)-YY(IJM-NJ)+2.0*STAG)	
488	C	
489	PHISE=PHINE	
490	PHINE=PHI(IJ)*FYS+PHI(IJP)*FYN+FXW	
491	& (PHI(IJ)*FYS+PHI(IPJ+1)*FYN)*FXE	
492	SEWS=SEWN	
493	SEWN=GAMN/AREN*(DXN*DXET+DYN*DYET)*(PHINE-PHINW(J))	
494	SNSE=-GAME/AREE*(DXKS+DXE+DYKS+DYE)*(PHINE-PHISE)	
495	IF (I.EQ.NIM.AND.(JTBE(J).EQ.3.OR.JTBE(J).EQ.4)) SNSE=0.	
496	IF (J.EQ.NJM.AND.(ITBN(I).EQ.3.OR.ITBN(I).EQ.4)) SEWN=0.	
497	APV(IJ)=SNSE-SNSW(J)+SEWN-SEWS	

Oct 12 1996 16:38	mskmod_2d	Page 8
498	C	
499	C LINEAR UPWIND DIFFERENCING	
500	C	
501	IMJ1=IMJ-NJ	
502	IMJ2=MAX(1,IMJ1)	
503	APV(IJ)=APV(IJ)+ARE*PHI(IPJ+NJ)+AWW*PHI(IMJ2)+ANN	
504	& PHI(IJP+1)+ASS*PHI(IJM-1)	
505	& AE1*PHI(IPJ)+AW1*PHI(IMJ)+AN1*PHI(IJP)+AS1*PHI(IJM)	
506	APU(IJ)=ARE*AWN+ANN+ASS+AE1+AW1+AN1+AS1	
507	C	
508	VISS=VIS(IJ)-VISCOS+SMALL	
509	C	
510	GO TO (120,121,130,131) IPHI	
511	C	
512	C-----TURBULENT KINETIC ENERGY SOURCE TERMS	
513	C	
514	120 CONTINUE	
515	DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))	
516	DYNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))	
517	DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))	
518	DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))	
519	RP=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IMJ-1))	
520	US=UN	
521	VS=VN	
522	WS=WN	
523	UN=U(IJP)*FYN+U(IJ)*FYS	
524	VN=V(IJP)*FYN+V(IJ)*FYS	
525	WN=W(IJP)*FYN+W(IJ)*FYS	
526	UE=U(IPJ)*FXE+U(IJ)*FXW	
527	VE=V(IPJ)*FXE+V(IJ)*FXW	
528	WE=W(IPJ)*FXE+W(IJ)*FXW	
529	DUEW=UE-UM(J)	
530	DUNS=UN-US	
531	DVEW=VE-VM(J)	
532	DVNS=VN-VS	
533	DNEW=WE-WW(J)	
534	DVNS=WN-WS	
535	GTERM=(2.0*((DUEW*DUNS-DVNS*DVEW)**2)+(DUNS*DNEW-DUEW*DVNS+DVEW*DVNS-DVNS*DVEW)**2)/(ARE(IJ)**2)	
537	IF (AKSI)	
538	& GTERM=GTERM+(DUNS*DNEW-DVNS*DVNS)**2+	
539	& (DNEW*DUNS-DVNS*DVEW-UM(J)/RP)*ARE(IJ)**2)	
540	& (DNEW*DUNS-DVNS*DVEW-UM(J)/RP)*ARE(IJ)**2)	
541	& /ARE(IJ)**2)	
542	& +2.0*(V(IJ)/RP)**2	
543	C	
544	DIVUV=(DUEW*DUNS-DVNS*DVEW+DVNS*DNEW-DVNS*DNEW)/ARE(IJ)	
545	GTERM=GTERM-2./3.*DIVUV**2	
546	GTERM=MAX(GTERM,0.)	
547	C	
548	C GENERATION OF KINETIC ENERGY	
549	C	
550	GEN(IJ)=(VIS(IJ)-VISCOS)*GTERM	
551	SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)	
552	C	
553	BP(IJ)=APU(IJ)+ED(IJ)*DEN(IJ)*VOL(IJ)/(TE(IJ)+SMALL)	
554	C	
555	GO TO 150	
556	C	
557	C-----DISSIP. OF TURB. KIN. ENERGY SOURCE TERMS	
558	C	
559	130 CONTINUE	
560	C	
561	SU(IJ)=APV(IJ)+(CPI*GEN(IJ)**2/DEN(IJ)+CP2*GEN(IJ)*ED(IJ))*	
562	& VOL(IJ)/(TE(IJ)+SMALL)	
563	C	
564	BP(IJ)=APU(IJ)+CP3*DEN(IJ)*ED(IJ)*VOL(IJ)/(TE(IJ)+SMALL)	
565	C	
566	GO TO 150	
567	C	
568	C-----SMALLER SCALE KINETIC ENERGY	

Oct 12 1996 16:38

mskmod_2d

Page 7

Oct 12 1996 16:38

mskmod_2d

Page 8

Oct 12 1996 16:38	mskemod_2d	Page 9
569 121	CONTINUE	
570	SU(IJ)=APV(IJ)+(ED(IJ)-EDT(IJ))*DEN(IJ)*VOL(IJ)	
571	BP(IJ)=APU(IJ)	
572	GO TO 150	
573 C	SMALLER SCALE ENERGY DISSIPATION	
574 C---		
575 C	CONTINUE	
576 131	CONTINUE	
577 C	SU(IJ)=APV(IJ)+(CT1*ED(IJ)**2+CT2*ED(IJ)*EDT(IJ))*DEN(IJ)*	
578	VOL(IJ)/(TET(IJ)+SMALL)	
579	&	
580	BP(IJ)=APU(IJ)+DEN(IJ)*CT3*EDT(IJ)*VOL(IJ)/(TET(IJ)+SMALL)	
581 C		
582 150	CONTINUE	
583	UW(IJ)=UE	
584	VW(IJ)=VE	
585	WW(IJ)=WE	
586	SNW(IJ)=SNSF	
587	PHINW(IJ)=PHINE	
588	FYSS=FY(IJM)	
589	FXW(IJ)=FX(IMJ)	
590	DW(IJ)=DE	
591 C		
592 101	CONTINUE	
593 100	CONTINUE	
594 C	PROBLEM MODIFICATIONS - BOUNDARY CONDITIONS	
595 C		
596 C	IDIR=IPHI	
597	CALL MODPHI	
598		
599 C	IF (IPHI GE 3.AND.LAY2) THEN	
600	DO 300 I=2,NIM	
601	IF (ITBS(I).NE.4) GO TO 301	
602	DO 310 J=2,JZLS	
603	IJ=IMNJ(I)+J	
604	SU(IJ)=GREAT*ED(IJ)	
605	BP(IJ)=GREAT	
606	CONTINUE	
607 310	CONTINUE	
608 301	IF (ITBN(I).NE.4) GO TO 300	
609	DO 320 J=JZLN,NJM	
610	IJ=IMNJ(I)+J	
611	SU(IJ)=GREAT*ED(IJ)	
612	BP(IJ)=GREAT	
613 320	CONTINUE	
614 300	CONTINUE	
615 C		
616	DO 400 J=2,NJM	
617	IF (JTBW(J).NE.4) GO TO 401	
618	DO 410 I=2,I2LW	
619	IJ=IMNJ(I)+J	
620	SU(IJ)=GREAT*ED(IJ)	
621	BP(IJ)=GREAT	
622 410	CONTINUE	
623 401	IF (JTBE(I).NE.4) GO TO 400	
624	DO 420 I=I2LE,NIM	
625	IJ=IMNJ(I)+J	
626	SU(IJ)=GREAT*ED(IJ)	
627	BP(IJ)=GREAT	
628 420	CONTINUE	
629 400	CONTINUE	
630	ENDIF	
631 C		
632	DO 200 I=2,NIM	
633	DO 201 J=2,NJM	
634	IJ=IMNJ(I)+J	
635	AP(IJ)=AW(IJ)+AE(IJ)+AN(IJ)+AS(IJ)+BP(IJ)	
636	AP(IJ)=AP(IJ)/URFPHI	
637	SU(IJ)=SU(IJ)+(1.-URFPHI)*AP(IJ)*PHI(IJ)	
638	201 CONTINUE	
639	200 CONTINUE	

Oct 12 1996 16:38	mskemod_2d	Page 10
640 C		
641 C	SOLVING F.D. EQUATIONS	
642 C		
643	CALL SOLSIP (PHI, IPHI)	
644 C		
645	RETURN	
646	END	
647 C		
648 C		
649	SUBROUTINE TWOLAY	
650 C		
651	INCLUDE 'mskemod.h'	
652 C		
653	DATA CMU25,CMU75,CAPPA,ELOG/0.5477,0.1643,0.4197,9.0/	
654 C		
655	C11=CAPPA/CMU75	
656	AED=2.0*C11	
657	AMU=70.0	
658 C		
659 C	ALONG THE SOUTH BOUNDARY	
660 C		
661	DO 70 I=2,NIM	
662	IF (ITBS(I).NE.4) GO TO 70	
663	DISNS=GREAT	
664	DISNW=GREAT	
665	DISNE=GREAT	
666 C		
667	DO 71 J=2,JZLS	
668	IJ=IMNJ(I)+J	
669	IJW=IMNJ(I)+1	
670	IMJW=IJW-NJ	
671	DXB=XX(IJW)-XX(IMJW)	
672	DYB=YY(IJW)-YY(IMJW)	
673	XPW=HAF*(XX(IJW)+XX(IMJW))	
674	YPW=HAF*(YY(IJW)+YY(IMJW))	
675	XPB=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
676	YPB=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
677	DXBP=XBP-XPW	
678	DYBP=YBP-YPW	
679	DISNS=DELTA(DXB,DYB,DXBP,DYBP)	
680	C--CHECK WEST BOUNDARY	
681	IF (JTBW(J).EQ.4) THEN	
682	IJW=J	
683	IMJW=IJW-1	
684	DXB=XX(IJW)-XX(IMJW)	
685	DYB=YY(IJW)-YY(IMJW)	
686	XB=HAF*(XX(IJW)+XX(IMJW))	
687	YB=HAF*(YY(IJW)+YY(IMJW))	
688	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
689	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
690	DXBP=XBP-XB	
691	DYBP=YBP-YB	
692	DISNW=DELTA(DXB,DYB,DXBP,DYBP)	
693	ENDIF	
694	C--CHECK EAST BOUNDARY	
695	IF (JTBE(J).EQ.4) THEN	
696	IJW=IMNJ(NIM)+J	
697	IMJW=IJW-1	
698	DXB=XX(IJW)-XX(IMJW)	
699	DYB=YY(IJW)-YY(IMJW)	
700	XB=HAF*(XX(IJW)+XX(IMJW))	
701	YB=HAF*(YY(IJW)+YY(IMJW))	
702	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
703	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
704	DXBP=XBP-XB	
705	DYBP=YBP-YB	
706	DISNE=DELTA(DXB,DYB,DXBP,DYBP)	
707	ENDIF	
708 C		
709	DISN=MIN(DISNS,DISNW,DISNE)	
710	TETOT=TE(IJ)+TET(IJ)	

Oct 12 1996 16:38	mskmod_2d	Page 11
711	RK=DISN*DEN(IJ)*SQRT(TETOT)/VISCOS	
712	ALMU=C11*DISN*(1.0-EXP(-RK/AMU))	
713	ALSD=C11*DISN*(1.0-EXP(-RK/AED))	
714	ED(IJ)=TETOT*SQRT(TETOT)/ALED	
715	VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TETOT)*ALMU	
716 71	CONTINUE	
717 70	CONTINUE	
718 C	...ALONG THE NORTH BOUNDARY	
719 C		
720 C		
721	DO 72 I=2,NJM	
722	IF (ITBN(I).NE.4) GO TO 72	
723	DISNN=GREAT	
724	DISNW=GREAT	
725	DISNE=GREAT	
726	DO 73 J=I2LN,NJM	
727	IJ=IMNJ(I)+J	
728	IMJW=IMNJ(I)+NJM	
729	IMJW=IJW-NJ	
730	DXB=XX(IJW)-XX(IMJW)	
731	DYB=YY(IJW)-YY(IMJW)	
732	XB=HAF*(XX(IJW)+XX(IMJW))	
733	YB=HAF*(YY(IJW)+YY(IMJW))	
734	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
735	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
736	DXBP=XBP-XB	
737	DYBP=YBP-YB	
738	DISNN=DELTA(DXB,DYB,DXBP,DYBP)	
739 C	--CHECK WEST BOUNDARY	
740	IF (JTBM(J).EQ.4) THEN	
741	IJW=J	
742	IMJW=IJW-1	
743	DXB=XX(IJW)-XX(IMJW)	
744	DYB=YY(IJW)-YY(IMJW)	
745	XB=HAF*(XX(IJW)+XX(IMJW))	
746	YB=HAF*(YY(IJW)+YY(IMJW))	
747	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
748	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
749	DXBP=XBP-XB	
750	DYBP=YBP-YB	
751	DISNW=DELTA(DXB,DYB,DXBP,DYBP)	
752	ENDIF	
753 C	--CHECK EAST BOUNDARY	
754	IF (JTBE(J).EQ.4) THEN	
755	IJW=IMNJ(NIM)+J	
756	IMJW=IJW-1	
757	DXB=XX(IJW)-XX(IMJW)	
758	DYB=YY(IJW)-YY(IMJW)	
759	XB=HAF*(XX(IJW)+XX(IMJW))	
760	YB=HAF*(YY(IJW)+YY(IMJW))	
761	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
762	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
763	DXBP=XBP-XB	
764	DYBP=YBP-YB	
765	DISNE=DELTA(DXB,DYB,DXBP,DYBP)	
766	ENDIF	
767 C		
768	DISN=MIN(DISNN,DISNW,DISNE)	
769	TETOT=TE(IJ)+TET(IJ)	
770	RK=DISN*DEN(IJ)*SQRT(TETOT)/VISCOS	
771	ALMU=C11*DISN*(1.0-EXP(-RK/AMU))	
772	ALSD=C11*DISN*(1.0-EXP(-RK/AED))	
773	ED(IJ)=TETOT*SQRT(TETOT)/ALED	
774	VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TETOT)*ALMU	
775 73	CONTINUE	
776 72	CONTINUE	
777 C		
778 C		
779 C	...ALONG THE WEST BOUNDARY	
780 C		
781	DO 75 J=2,NJM	

Oct 12 1996 16:38	mskmod_2d	Page 12
782	IF (JTBM(J).NE.4) GO TO 75	
783	DISNW=GREAT	
784	DISNN=GREAT	
785	DISNS=GREAT	
786	DO 76 I=2,I2LW	
787	IJ=IMNJ(I)+J	
788	IJW=J	
789	IMJW=IJW-1	
790	DXB=XX(IJW)-XX(IMJW)	
791	DYB=YY(IJW)-YY(IMJW)	
792	XB=HAF*(XX(IJW)+XX(IMJW))	
793	YB=HAF*(YY(IJW)+YY(IMJW))	
794	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
795	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
796	DXBP=XBP-XB	
797	DYBP=YBP-YB	
798	DISNW=DELTA(DXB,DYB,DXBP,DYBP)	
799 C	--CHECK NORTH BOUNDARY	
800	IF (ITBN(I).EQ.4) THEN	
801	IJW=IMNJ(I)+NJM	
802	IMJW=IJW-NJ	
803	DXB=XX(IJW)-XX(IMJW)	
804	DYB=YY(IJW)-YY(IMJW)	
805	XB=HAF*(XX(IJW)+XX(IMJW))	
806	YB=HAF*(YY(IJW)+YY(IMJW))	
807	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
808	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
809	DXBP=XBP-XB	
810	DYBP=YBP-YB	
811	DISNN=DELTA(DXB,DYB,DXBP,DYBP)	
812	ENDIF	
813 C	--CHECK SOUTH BOUNDARY	
814	IF (ITBS(I).EQ.4) THEN	
815	IJW=IMNJ(I)+1	
816	IMJW=IJW-NJ	
817	DXB=XX(IJW)-XX(IMJW)	
818	DYB=YY(IJW)-YY(IMJW)	
819	XPW=HAF*(XX(IJW)+XX(IMJW))	
820	YPW=HAF*(YY(IJW)+YY(IMJW))	
821	XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))	
822	YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))	
823	DXBP=XBP-XPW	
824	DYBP=YBP-YPW	
825	DISNS=DELTA(DXB,DYB,DXBP,DYBP)	
826	ENDIF	
827 C		
828	DISN=MIN(DISNW,DISNS,DISNN)	
829	TETOT=TE(IJ)+TET(IJ)	
830	RK=DISN*DEN(IJ)*SQRT(TETOT)/VISCOS	
831	ALMU=C11*DISN*(1.0-EXP(-RK/AMU))	
832	ALSD=C11*DISN*(1.0-EXP(-RK/AED))	
833	ED(IJ)=TETOT*SQRT(TETOT)/ALED	
834	VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TETOT)*ALMU	
835 76	CONTINUE	
836 75	CONTINUE	
837 C		
838 C	...ALONG THE EAST BOUNDARY	
839 C		
840	DO 78 J=2,NJM	
841	IF (JTBE(J).NE.4) GO TO 78	
842	DISNE=GREAT	
843	DISNN=GREAT	
844	DISNS=GREAT	
845	DO 79 I=I2LE,NIM	
846	IJ=IMNJ(I)+J	
847	IJW=IMNJ(NIM)+J	
848	IMJW=IJW-1	
849	DXB=XX(IJW)-XX(IMJW)	
850	DYB=YY(IJW)-YY(IMJW)	
851	XB=HAF*(XX(IJW)+XX(IMJW))	
852	YB=HAF*(YY(IJW)+YY(IMJW))	

Oct 12 1996 16:38 mskemod_2d Page 13

```

853 XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
854 YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
855 DXBP=XBP-XB
856 DYBP=YBP-YB
857 DISN=DELTA(DXB,DYB,DXBP,DYBP)
858 C--CHECK NORTH BOUNDARY
859 IF(ITBN(I),EQ.4) THEN
860 IJW=IMNJ(I)+NJW
861 IJW=IJW-NJ
862 DXB=XX(IJW)-XX(IMJW)
863 DYB=YY(IJW)-YY(IMJW)
864 XB=HAF*(XX(IJW)+XX(IMJW))
865 YB=HAF*(YY(IJW)+YY(IMJW))
866 XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
867 YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
868 DXBP=XBP-XB
869 DYBP=YBP-YB
870 DISN=DELTA(DXB,DYB,DXBP,DYBP)
871 ENDIF
872 C--CHECK SOUTH BOUNDARY
873 IF(ITBS(I),EQ.4) THEN
874 IJW=IMNJ(I)+1
875 IJW=IJW-NJ
876 DXB=XX(IJW)-XX(IMJW)
877 DYB=YY(IJW)-YY(IMJW)
878 XPW=HAF*(XX(IJW)+XX(IMJW))
879 YPW=HAF*(YY(IJW)+YY(IMJW))
880 XBP=QTR*(XX(IJ)+XX(IJ-1)+XX(IJ-NJ-1)+XX(IJ-NJ))
881 YBP=QTR*(YY(IJ)+YY(IJ-1)+YY(IJ-NJ-1)+YY(IJ-NJ))
882 DXBP=XBP-XPW
883 DYBP=YBP-YPW
884 DISN=DELTA(DXB,DYB,DXBP,DYBP)
885 ENDIF
886 C
887 DISN=MIN(DISN,DISNS,DISNN)
888 TETOT=TE(IJ)+TET(IJ)
889 RK=DISN*DEN(IJ)*SORT(TETOT)/VISCOS
890 ALMU=C11*DISN*(1.0-EXP(-RK/ALMU))
891 ALED=C11*DISN*(1.0-EXP(-RK/ALED))
892 ED(IJ)=TETOT*QTR(TETOT)/ALED
893 VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SORT(TETOT)*ALMU
894 79 CONTINUE
895 78 CONTINUE
896 C
897 RETURN
898 END
899 C
900 FUNCTION DELTA(DXB,DYB,DXBP,DYBP)
901 ARW=SQRT(DXB**2+DYB**2)
902 DXB=DXB/ARW
903 DYB=DYB/ARW
904 DELP=DXB*DXBP+DYB*DYBP
905 DELTA=SQRT(DXB**2+DYBP**2-DELP**2)
906 RETURN
907 END
908 C
909 C
910 SUBROUTINE SOLSIP(PHI,IPHI)
911 C
912 INCLUDE 'mskemod.h'
913 C
914 DIMENSION PHI(NXNY)
915 ID=IPHI
916 L=0
917 DO 5 I=2,NIM
918 DO 5 J=2,NJM
919 IJ=IMNJ(I)+J
920 API=1.0/AP(IJ)
921 AP(IJ)=1.0
922 AE(IJ)=AE(IJ)*API
923 AW(IJ)=AW(IJ)*API

```

Oct 12 1996 16:38 mskemod_2d Page 14

```

924 AN(IJ)=AN(IJ)*API
925 AS(IJ)=AS(IJ)*API
926 SU(IJ)=SU(IJ)*API
927 5 CONTINUE
928 C
929 DO 10 I=2,NIM
930 DO 11 J=2,NJM
931 IJ=IMNJ(I)+J
932 IJW=IJ-1
933 IMJ=IJ-NJ
934 BW(IJ)=-AW(IJ)/(1.+ALFA*BN(IMJ))
935 BS(IJ)=-AS(IJ)/(1.+ALFA*BE(IJM))
936 POM1=ALFA*BW(IJ)*BN(IMJ)
937 POM2=ALFA*BS(IJ)*BE(IJM)
938 BP(IJ)=AP(IJ)+POM1+POM2-BW(IJ)*BE(IMJ)-BS(IJ)*BN(IJM)
939 BN(IJ)=(-AN(IJ)+POM1)/(BP(IJ)+SMALL)
940 BE(IJ)=(-AE(IJ)+POM2)/(BP(IJ)+SMALL)
941 11 CONTINUE
942 10 CONTINUE
943 NSTP=NSWP(ID)
944 100 CONTINUE
945 L=L+1
946 RESORP=0.
947 DO 20 I=2,NIM
948 DO 21 J=2,NJM
949 IJ=IMNJ(I)+J
950 RES(IJ)=AN(IJ)*PHI(IJ+1)+AS(IJ)*PHI(IJ-1)+AE(IJ)*PHI(IJ+NJ)+
& AW(IJ)*PHI(IJ-NJ)+SU(IJ)*AP(IJ)*PHI(IJ)
951 RESORP=RESORP+ABS(RES(IJ))
952 RES(IJ)=(RES(IJ)-BS(IJ)*RES(IJ-1)-BW(IJ)*RES(IJ-NJ))/
& (BP(IJ)+SMALL)
953 21 CONTINUE
954 20 CONTINUE
955 IF(L.EQ.1) RESOR(ID)=RESORP
956 IF(L.EQ.1) RESOR(ID)=RESORP
957 RSM=SOR(ID)*RESOR(ID)
958 DO 30 I=2,NIM
959 IJ=IMNJ(I)+J
960 IJ=IMNJ(I)+J
961 DO 31 J=2,NJM
962 JJ=NMJ+2-J
963 IJ=IMNJ(IJ)+JJ
964 RES(IJ)=RES(IJ)-BN(IJ)*RES(IJ+1)-BE(IJ)*RES(IJ+NJ)
965 PHI(IJ)=PHI(IJ)+RES(IJ)
966 31 CONTINUE
967 30 CONTINUE
968 IF(TEST) PRINT 1, L, RESORP
969 IF(RESORP.GT.RSM.AND.L.LT.NSTP) GO TO 100
970 IF(RESORP.GT.RSM.AND.L.GE.NSTP) WRITE(6,2)
971 1 FORMAT(10X,15,' SWEEP, RESOR =',E12.4)
972 2 FORMAT(//,10X,' SIPSOL DID NOT CONVERGE ',//)
973 RETURN
974 END
975 C
976 C
977 SUBROUTINE USERM
978 C
979 INCLUDE 'mskemod.h'
980 C
981 C
982 DATA CAPPA/0.4197/
983 C
984 C
985 ENTRY MODVIS
986 C
987 DO 80 I=1,NI
988 DO 80 J=1,NJ
989 IJ=IMNJ(I)+J
990 VISOLD=VIS(IJ)
991 VIS(IJ)=VISCOS
992 IF(ED(IJ).GT.SMALL)
& VIS(IJ)=DEN(IJ)*TE(IJ)+TET(IJ)*TE(IJ)*CMU/ED(IJ)+VISOLD
993 VIS(IJ)=URFVIS*VIS(IJ)+(1.-URFVIS)*VISOLD
994

```

Oct 12 1996 16:38		mskmod_2d	Page 15
995	80	CONTINUE	
996	C	IF (LAY2) THEN	
997		DO 81 I=2,NIM	
998		IF (ITBS(I).NE.4) GO TO 82	
999		DO 83 J=2,J2LS	
1000		IJ=IMNJ(I)+J	
1001		VIS(IJ)=VIS2(IJ)	
1002		CONTINUE	
1003	83	CONTINUE	
1004	82	IF (ITBN(I).NE.4) GO TO 81	
1005		DO 84 J=J2LN,NJM	
1006		IJ=IMNJ(I)+J	
1007		VIS(IJ)=VIS2(IJ)	
1008	84	CONTINUE	
1009	81	CONTINUE	
1010		DO 85 J=2,NJM	
1011		IF (JTBE(J).NE.4) GO TO 86	
1012		DO 87 I=I2LE,NIM	
1013		IJ=IMNJ(I)+J	
1014		VIS(IJ)=VIS2(IJ)	
1015	87	CONTINUE	
1016	86	IF (JTEW(J).NE.4) GO TO 85	
1017		DO 88 I=2,I2LW	
1018		IJ=IMNJ(I)+J	
1019		VIS(IJ)=VIS2(IJ)	
1020	88	CONTINUE	
1021	85	CONTINUE	
1022	C	ENDIF	
1023		RETURN	
1024	C		
1025			
1026			
1027	C	ENTRY MODPHI	
1028			
1029	C	GO TO (800,900,1000,1100) IDIR	
1030			
1031	C	-----BOUNDARY CONDITIONS FOR TURBULENT KINETIC ENERGY (KP)	
1032			
1033	C		
1034	800	CONTINUE	
1035	C	-----SOUTH BOUNDARY	
1036		DO 810 I=2,NIM	
1037		IJ=IMNJ(I)+2	
1038		GO TO (811,812,813,814) ITBS(I)	
1039	811	CONTINUE	
1040		SU(IJ)=SU(IJ)+AS(IJ)*TE(IJ-1)	
1041		BP(IJ)=BP(IJ)+AS(IJ)	
1042		GO TO 815	
1043	812	TE(IJ-1)=TE(IJ)	
1044		GO TO 815	
1045	813	CONTINUE	
1046		IJ=IJ-1	
1047		IPJ=IJ-NJ	
1048		IMJ=IJ-NJ	
1049		FXE1=FX(IJ)	
1050		FXE2=FX(IMJ)	
1051		FXW1=1.-FXE1	
1052		FXW2=1.-FXE2	
1053		DXB=XX(IJ)-XX(IMJ)	
1054		DYB=YY(IJ)-YY(IMJ)	
1055		DXBP=QTR*(XX(IJ+1)-XX(IMJ+1))	
1056		DYBP=QTR*(YY(IJ+1)-YY(IMJ+1))	
1057		FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)	
1058		DEL=TE(IJ)*(FXW1-FXE2)+TE(IPJ)*FXE1-TE(IMJ)*FXW2	
1059		TE(IJ)=TE(IJ+1)-DEL*FAC	
1060		IJ=IJ+1	
1061		GO TO 815	
1062	814	CONTINUE	
1063		IF (.NOT.LAY2) GEN(IJ)=GENTS(I)	
1064		RP=QTR*(R(IJ)+R(IJ-NO)+R(IJ-1)+R(IJ-NJ-1))	
1065		IF (AKSI) THEN	

Oct 12 1996 16:38		mskmod_2d	Page 16
1066		IMJ=IJ-NJ	
1067		IQM=IQ-1	
1068		IJP=IJ+1	
1069		IPJ=IJ+NJ	
1070		FYN=FY(IJ)	
1071		FYS=1.0-FYN	
1072		FXE=FX(IJ)	
1073		FXW=1.0-FXE	
1074		DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IQM)-XX(IMJ-1))	
1075		DXNS=HAF*(XX(IJ)-XX(IQM)+XX(IMJ)-XX(IMJ-1))	
1076		DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IQM)-YY(IMJ-1))	
1077		DYNS=HAF*(YY(IJ)-YY(IQM)+YY(IMJ)-YY(IMJ-1))	
1078		WN=W(IJP)*FYN+W(IJ)*FYS	
1079		WS=W(IJ)*FY(IJ-1)+W(IJ-1)*(1.0-FY(IJ-1))	
1080		WE=W(IPJ)*FXE+W(IJ)*FXW	
1081		WW=W(IJ)*FX(IJ-NJ)+W(IJ-NJ)*(1.0-FX(IJ-NJ))	
1082		DWEW=WE-WW	
1083		DWNS=WN-WS	
1084		GEN(IJ)=GEN(IJ)+((DYNW*DWEW-DYEW*DWNS)**2+	
1085		& (DXEW*DWNS-DXNS*DWEW-(W(IJ)/RP)*ARE(IJ))**2)	
1086		& / (ARE(IJ)**2)	
1087		& +2.*(V(IJ)/RP)**2)*(VIS(IJ)-VISCOS)	
1088		ENDIF	
1089		SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)	
1090	815	CONTINUE	
1091		AS(IJ)=0.0	
1092	810	CONTINUE	
1093	C		
1094	C	-----NORTH BOUNDARY	
1095		DO 820 I=2,NIM	
1096		IJ=IMNJ(I)+NJM	
1097		GO TO (821,822,823,824) ITBN(I)	
1098	821	CONTINUE	
1099		SU(IJ)=SU(IJ)+AN(IJ)*TE(IJ+1)	
1100		BP(IJ)=BP(IJ)+AN(IJ)	
1101		GO TO 825	
1102	822	TE(IJ+1)=TE(IJ)	
1103		GO TO 825	
1104	823	CONTINUE	
1105		IJ=IJ+1	
1106		IPJ=IJ+NJ	
1107		IMJ=IJ-NJ	
1108		FXE1=FX(IJ)	
1109		FXE2=FX(IMJ)	
1110		FXW1=1.-FXE1	
1111		FXW2=1.-FXE2	
1112		DXB=XX(IJ)-XX(IMJ)	
1113		DYB=YY(IJ)-YY(IMJ)	
1114		DXBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))	
1115		DYBP=QTR*(YY(IJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))	
1116		FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)	
1117		DEL=TE(IJ)*(FXW1-FXE2)+TE(IPJ)*FXE1-TE(IMJ)*FXW2	
1118		TE(IJ)=TE(IJ-1)-DEL*FAC	
1119		IJ=IJ-1	
1120		GO TO 825	
1121	824	CONTINUE	
1122		IF (.NOT.LAY2) GEN(IJ)=GENTN(I)	
1123		RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))	
1124		IF (AKSI) THEN	
1125		IMJ=IJ-NJ	
1126		IJM=IJ-1	
1127		IJP=IJ+1	
1128		IPJ=IJ+NJ	
1129		FYN=FY(IJ)	
1130		FYS=1.0-FYN	
1131		FXE=FX(IJ)	
1132		FXW=1.0-FXE	
1133		DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IQM)-XX(IMJ-1))	
1134		DXNS=HAF*(XX(IJ)-XX(IQM)+XX(IMJ)-XX(IMJ-1))	
1135		DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IQM)-YY(IMJ-1))	
1136		DYNS=HAF*(YY(IJ)-YY(IQM)+YY(IMJ)-YY(IMJ-1))	

Oct 12 1996 16:38	mskemod_2d	Page 19
1279	SU(IJ)=SU(IJ)+AS(IJ)*TET(IJ-1)	
1280	BP(IJ)=BP(IJ)+AS(IJ)	
1281	GO TO 915	
1282	TET(IJ-1)=TET(IJ)	
1283	GO TO 915	
1284	CONTINUE	
1285	IJ=IJ-1	
1286	IFJ=IJ-NJ	
1287	IMJ=IJ-NJ	
1288	FXE1=FX(IJ)	
1289	FXE2=FX(IMJ)	
1290	FXW1=1.-FXE1	
1291	FXW2=1.-FXE2	
1292	DXB=XX(IJ)-XX(IMJ)	
1293	DYB=YY(IJ)-YY(IMJ)	
1294	DXBP=QTR*(XX(IJ+1)-XX(IMJ+1)-XX(IMJ))	
1295	DYBP=QTR*(YY(IJ+1)-YY(IMJ+1)-YY(IMJ))	
1296	FAC=(DXB+DYB+DYB*DYB)/(DXB**2+DYB**2+SMALL)	
1297	DEL=TET(IJ)*(FXW1-FXE2)+TET(IFJ)*FXE1-TET(IMJ)*FXW2	
1298	TET(IJ)=TET(IJ+1)-DEL*FAC	
1299	IJ=IJ+1	
1300	GO TO 915	
1301	CONTINUE	
1302	SU(IJ)=FRACTW*TE(IJ)*GREAT	
1303	BP(IJ)=GREAT	
1304	CONTINUE	
1305	AS(IJ)=0.0	
1306	CONTINUE	
1307	C-----NORTH BOUNDARY	
1308	DO 920 I=2,NIM	
1309	IJ=IMNJ(I)+NJM	
1310	GO TO (921,922,923,924) ITBN(I)	
1311	CONTINUE	
1312	SU(IJ)=SU(IJ)+AN(IJ)*TET(IJ+1)	
1313	BP(IJ)=BP(IJ)+AN(IJ)	
1314	GO TO 925	
1315	922 TET(IJ+1)=TET(IJ)	
1316	GO TO 925	
1317	923 CONTINUE	
1318	IJ=IJ+1	
1319	IFJ=IJ+NJ	
1320	IMJ=IJ-NJ	
1321	FXE1=FX(IJ)	
1322	FXE2=FX(IMJ)	
1323	FXW1=1.-FXE1	
1324	FXW2=1.-FXE2	
1325	DXB=XX(IJ)-XX(IMJ)	
1326	DYB=YY(IJ)-YY(IMJ)	
1327	DXBP=QTR*(XX(IJ-2)-XX(IMJ-2)-XX(IMJ))	
1328	DYBP=QTR*(YY(IJ-2)-YY(IMJ-2)-YY(IMJ))	
1329	FAC=(DXB+DYB+DYB*DYB)/(DXB**2+DYB**2+SMALL)	
1330	DEL=TET(IJ)*(FXW1-FXE2)+TET(IFJ)*FXE1-TET(IMJ)*FXW2	
1331	TET(IJ)=TET(IJ-1)-DEL*FAC	
1332	IJ=IJ-1	
1333	GO TO 925	
1334	CONTINUE	
1335	SU(IJ)=FRACTW*TE(IJ)*GREAT	
1336	BP(IJ)=GREAT	
1337	CONTINUE	
1338	AN(IJ)=0.0	
1339	CONTINUE	
1340	C-----WEST BOUNDARY	
1341	DO 930 J=2,NJM	
1342	IJ=IMNJ(2)+J	
1343	GO TO (931,932,933,934) JTBW(J)	
1344	CONTINUE	
1345	SU(IJ)=SU(IJ)+AW(IJ)*TET(IJ-NJ)	
1346	BP(IJ)=BP(IJ)+AW(IJ)	
1347	GO TO 935	
1348	932 TET(IJ-NJ)=TET(IJ)	
1349	GO TO 935	

Oct 12 1996 16:38	mskemod_2d	Page 20
1350	933 CONTINUE	
1351	IJ=J	
1352	IJP=IJ+1	
1353	IJM=IJ-1	
1354	FYN1=FY(IJ)	
1355	FYN2=FY(IJM)	
1356	FYS1=1.-FYN1	
1357	FYS2=1.-FYN2	
1358	DYB=YY(IJ)-YY(IJM)	
1359	DXB=XX(IJ)-XX(IJM)	
1360	DXBP=QTR*(XX(IJ+NJ)-XX(IJ)-XX(IJM+NJ)-XX(IJM))	
1361	DYBP=QTR*(YY(IJ+NJ)-YY(IJ)-YY(IJM+NJ)-YY(IJM))	
1362	FAC=(DXB+DYB+DYB*DYB)/(DXB**2+DYB**2+SMALL)	
1363	DEL=TET(IJ)*(FYS1-FYN2)+TET(IJP)*FYN1-TET(IJM)*FYS2	
1364	TET(IJ)=TET(IJ+NJ)-DEL*FAC	
1365	IJ=IJ+NJ	
1366	GO TO 935	
1367	CONTINUE	
1368	SU(IJ)=FRACTW*TE(IJ)*GREAT	
1369	BP(IJ)=GREAT	
1370	CONTINUE	
1371	AW(IJ)=0.0	
1372	CONTINUE	
1373	C-----EAST BOUNDARY	
1374	DO 940 J=2,NJM	
1375	IJ=IMNJ(NIM)+J	
1376	GO TO (941,942,943,944) JTBE(J)	
1377	CONTINUE	
1378	SU(IJ)=SU(IJ)+AE(IJ)*TET(IJ+NJ)	
1379	BP(IJ)=BP(IJ)+AE(IJ)	
1380	GO TO 945	
1381	942 TET(IJ+NJ)=TET(IJ)	
1382	GO TO 945	
1383	CONTINUE	
1384	IJ=IJ+NJ	
1385	IJP=IJ+1	
1386	IJM=IJ-1	
1387	FYN1=FY(IJ)	
1388	FYN2=FY(IJM)	
1389	FYS1=1.-FYN1	
1390	FYS2=1.-FYN2	
1391	DXB=XX(IJ)-XX(IJM)	
1392	DYB=YY(IJ)-YY(IJM)	
1393	DXBP=QTR*(XX(IJ-NJ-NJ)-XX(IJ)+XX(IJM-NJ-NJ)-XX(IJM))	
1394	DYBP=QTR*(YY(IJ-NJ-NJ)-YY(IJ)+YY(IJM-NJ-NJ)-YY(IJM))	
1395	FAC=(DXB+DYB+DYB*DYB)/(DXB**2+DYB**2+SMALL)	
1396	DEL=TET(IJ)*(FYS1-FYN2)+TET(IJP)*FYN1-TET(IJM)*FYS2	
1397	TET(IJ)=TET(IJ-NJ)-DEL*FAC	
1398	IJ=IJ-NJ	
1399	GO TO 945	
1400	CONTINUE	
1401	SU(IJ)=FRACTW*TE(IJ)*GREAT	
1402	BP(IJ)=GREAT	
1403	CONTINUE	
1404	AE(IJ)=0.0	
1405	CONTINUE	
1406	C	
1407	RETURN	
1408	C	
1409	C-----BOUNDARY CONDITIONS FOR ENERGY TRANSFER RATE (ED).	
1410	C	
1411	1000 CONTINUE	
1412	C	
1413	CMU25=SORT(SORT(CMU))	
1414	CMU75=CMU25**3	
1415	C	
1416	C-----SOUTH BOUNDARY	
1417	DO 1010 I=2,NIM	
1418	IJ=IMNJ(I)+2	
1419	GO TO (1011,1012,1013,1014) ITBS(I)	
1420	CONTINUE	

Oct 12 1996 16:38 mskemod_2d Page 21

```

1421 SU(IJ)=SU(IJ)+AS(IJ)*ED(IJ-1)
1422 BP(IJ)=BP(IJ)+AS(IJ)
1423 GO TO 1015
1424 1012 ED(IJ-NJ)=ED(IJ)
1425 GO TO 1015
1426 1013 CONTINUE
1427 IJ=IJ-1
1428 IMJ=IJ+NJ
1429 IMJ=IJ+NJ
1430 FXE1=FX(IJ)
1431 FXE2=FX(IMJ)
1432 FXW1=1.-FXE1
1433 FXW2=1.-FXE2
1434 DXB=XX(IJ)-XX(IMJ)
1435 DYB=YY(IJ)-YY(IMJ)
1436 DXBP=QTR*(XX(IJ+1)-XX(IMJ+1)-XX(IMJ))
1437 DYBP=QTR*(YY(IJ+1)-YY(IMJ+1)-YY(IMJ))
1438 FAC=(DXB+DXBP+DYB+DYBP)/(DXB**2+DYB**2+SMALL)
1439 DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
1440 ED(IJ)=ED(IJ+1)-DEL*FAC
1441 IJ=IJ+1
1442 GO TO 1015
1443 1014 CONTINUE
1444 TT=ABS(TE(IJ)+TET(IJ))
1445 SU(IJ)=GREAT*CHU75*TT*SQRT(TT)/(CAPPA*DNN(I))
1446 BP(IJ)=GREAT
1447 1015 CONTINUE
1448 AS(IJ)=0.0
1449 1010 CONTINUE
1450 C-----NORTH BOUNDARY
1451 DO 1020 I=2,NJM
1452 IJ=IMNJ(I)+NJM
1453 GO TO (1021,1022,1023,1024) ITBN(I)
1454 1021 CONTINUE
1455 SU(IJ)=SU(IJ)+AN(IJ)*ED(IJ+1)
1456 BP(IJ)=BP(IJ)+AN(IJ)
1457 GO TO 1025
1458 1022 ED(IJ+1)=ED(IJ)
1459 GO TO 1025
1460 1023 CONTINUE
1461 IJ=IJ+1
1462 IPJ=IJ+NJ
1463 IMJ=IJ-NJ
1464 FXE1=FX(IJ)
1465 FXE2=FX(IMJ)
1466 FXW1=1.-FXE1
1467 FXW2=1.-FXE2
1468 DXB=XX(IJ)-XX(IMJ)
1469 DYB=YY(IJ)-YY(IMJ)
1470 DXBP=QTR*(XX(IJ-2)-XX(IJ)+XX(IMJ-2)-XX(IMJ))
1471 DYBP=QTR*(YY(IJ-2)-YY(IJ)+YY(IMJ-2)-YY(IMJ))
1472 FAC=(DXB+DXBP+DYB+DYBP)/(DXB**2+DYB**2+SMALL)
1473 DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2
1474 ED(IJ)=ED(IJ-1)-DEL*FAC
1475 IJ=IJ-1
1476 GO TO 1025
1477 1024 CONTINUE
1478 TT=ABS(TE(IJ)+TET(IJ))
1479 SU(IJ)=GREAT*CHU75*TT*SQRT(TT)/(CAPPA*DNN(I))
1480 BP(IJ)=GREAT
1481 1025 CONTINUE
1482 AN(IJ)=0.0
1483 1020 CONTINUE
1484 C-----WEST BOUNDARY
1485 DO 1030 J=2,NJM
1486 IJ=IMNJ(2)+J
1487 GO TO (1031,1032,1033,1034) JTBW(J)
1488 1031 CONTINUE
1489 SU(IJ)=SU(IJ)+AW(IJ)*ED(IJ-NJ)
1490 BP(IJ)=BP(IJ)+AW(IJ)
1491 GO TO 1035

```

Oct 12 1996 16:38 mskemod_2d Page 22

```

1492 1032 ED(IJ-NJ)=ED(IJ)
1493 GO TO 1035
1494 1033 CONTINUE
1495 IJ=J
1496 IJP=IJ+1
1497 IJM=IJ-1
1498 FYN1=FY(IJ)
1499 FYN2=FY(IMJ)
1500 FYS1=1.-FYN1
1501 FYS2=1.-FYN2
1502 DYB=YY(IJ)-YY(IMJ)
1503 DXB=XX(IJ)-XX(IMJ)
1504 DXBP=QTR*(XX(IJ+NJ)-XX(IJ)+XX(IMJ+NJ)-XX(IMJ))
1505 DYBP=QTR*(YY(IJ+NJ)-YY(IJ)+YY(IMJ+NJ)-YY(IMJ))
1506 FAC=(DXB+DXBP+DYB+DYBP)/(DXB**2+DYB**2+SMALL)
1507 DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IMJ)*FYS2
1508 ED(IJ)=ED(IJ+NJ)-DEL*FAC
1509 IJ=IJ+NJ
1510 GO TO 1035
1511 1034 CONTINUE
1512 TT=ABS(TE(IJ)+TET(IJ))
1513 SU(IJ)=GREAT*CHU75*TT*SQRT(TT)/(CAPPA*DNN(J))
1514 BP(IJ)=GREAT
1515 1035 CONTINUE
1516 AW(IJ)=0.0
1517 1030 CONTINUE
1518 C-----EAST BOUNDARY
1519 DO 1040 J=2,NJM
1520 IJ=IMNJ(NJM)+J
1521 GO TO (1041,1042,1043,1044) JTBE(J)
1522 1041 CONTINUE
1523 SU(IJ)=SU(IJ)+AE(IJ)*ED(IJ+NJ)
1524 BP(IJ)=BP(IJ)+AE(IJ)
1525 GO TO 1045
1526 1042 ED(IJ+NJ)=ED(IJ)
1527 GO TO 1045
1528 1043 CONTINUE
1529 IJ=IJ+NJ
1530 IJP=IJ+1
1531 IJM=IJ-1
1532 FYN1=FY(IJ)
1533 FYN2=FY(IMJ)
1534 FYS1=1.-FYN1
1535 FYS2=1.-FYN2
1536 DXB=XX(IJ)-XX(IMJ)
1537 DYB=YY(IJ)-YY(IMJ)
1538 DXBP=QTR*(XX(IJ-NJ)-XX(IJ)+XX(IMJ-NJ)-XX(IMJ))
1539 DYBP=QTR*(YY(IJ-NJ)-YY(IJ)+YY(IMJ-NJ)-YY(IMJ))
1540 FAC=(DXB+DXBP+DYB+DYBP)/(DXB**2+DYB**2+SMALL)
1541 DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IMJ)*FYS2
1542 ED(IJ)=ED(IJ-NJ)-DEL*FAC
1543 IJ=IJ-NJ
1544 GO TO 1045
1545 1044 CONTINUE
1546 TT=ABS(TE(IJ)+TET(IJ))
1547 SU(IJ)=GREAT*CHU75*TT*SQRT(TT)/(CAPPA*DNE(J))
1548 BP(IJ)=GREAT
1549 1045 CONTINUE
1550 AE(IJ)=0.0
1551 1040 CONTINUE
1552 C
1553 RETURN
1554 C
1555 C-----BOUNDARY CONDITIONS FOR ENERGY DISSIPATION RATE (EDT).
1556 C
1557 1100 CONTINUE
1558 C
1559 C-----SOUTH BOUNDARY
1560 DO 1110 I=2,NJM
1561 IJ=IMNJ(I)+2
1562 GO TO (1111,1112,1113,1114) ITBS(I)

```

Oct 12 1996 16:38	mskemod_2d	Page 23
1563	1111 CONTINUE	
1564	SU(IJ)=SU(IJ)+AS(IJ)*EDT(IJ-1)	
1565	BP(IJ)=BP(IJ)+AS(IJ)	
1566	GO TO 1115	
1567	1112 EDT(IJ-NJ)=EDT(IJ)	
1568	GO TO 1115	
1569	1113 CONTINUE	
1570	IJ=IJ-1	
1571	IPJ=IJ+NJ	
1572	IMJ=IJ-NJ	
1573	FXE1=FX(IJ)	
1574	FXE2=FX(IMJ)	
1575	FXW1=1.-FXE1	
1576	FXW2=1.-FXE2	
1577	DXB=XX(IJ)-XX(IMJ)	
1578	DYB=YY(IJ)-YY(IMJ)	
1579	DXBP=QTR*(XX(IJ+1)-XX(IMJ+1)-XX(IMJ))	
1580	DYBP=QTR*(YY(IJ+1)-YY(IMJ+1)-YY(IMJ))	
1581	FAC=(DXB*DXBP+DYB*DYP)/(DXB**2+DYB**2+SMALL)	
1582	DEL=EDT(IJ)*(FXW1-FXE2)+EDT(IPJ)*FXE1-EDT(IMJ)*FXW2	
1583	EDT(IJ)=EDT(IJ+1)-DEL*FAC	
1584	IJ=IJ+1	
1585	GO TO 1115	
1586	1114 CONTINUE	
1587	SU(IJ)=GREAT*ED(IJ)	
1588	BP(IJ)=GREAT	
1589	1115 CONTINUE	
1590	AS(IJ)=0.0	
1591	1110 CONTINUE	
1592	C-----NORTH BOUNDARY	
1593	DO 1120 I=2,NJM	
1594	IJ=IMNJ(I)-NJM	
1595	GO TO (1121,1122,1123,1124) ITBN(I)	
1596	1121 CONTINUE	
1597	SU(IJ)=SU(IJ)+AN(IJ)*EDT(IJ+1)	
1598	BP(IJ)=BP(IJ)+AN(IJ)	
1599	GO TO 1125	
1600	1122 EDT(IJ+1)=EDT(IJ)	
1601	GO TO 1125	
1602	1123 CONTINUE	
1603	IJ=IJ+1	
1604	IPJ=IJ+NJ	
1605	IMJ=IJ-NJ	
1606	FXE1=FX(IJ)	
1607	FXE2=FX(IMJ)	
1608	FXW1=1.-FXE1	
1609	FXW2=1.-FXE2	
1610	DXB=XX(IJ)-XX(IMJ)	
1611	DYB=YY(IJ)-YY(IMJ)	
1612	DXBP=QTR*(XX(IJ-2)-XX(IMJ-2)-XX(IMJ))	
1613	DYBP=QTR*(YY(IJ-2)-YY(IMJ-2)-YY(IMJ))	
1614	FAC=(DXB*DXBP+DYB*DYP)/(DXB**2+DYB**2+SMALL)	
1615	DEL=EDT(IJ)*(FXW1-FXE2)+EDT(IPJ)*FXE1-EDT(IMJ)*FXW2	
1616	EDT(IJ)=EDT(IJ-1)-DEL*FAC	
1617	IJ=IJ-1	
1618	GO TO 1125	
1619	1124 CONTINUE	
1620	SU(IJ)=GREAT*ED(IJ)	
1621	BP(IJ)=GREAT	
1622	1125 CONTINUE	
1623	AN(IJ)=0.0	
1624	1120 CONTINUE	
1625	C-----WEST BOUNDARY	
1626	DO 1130 J=2,NJM	
1627	IJ=IMNJ(2)+J	
1628	GO TO (1131,1132,1133,1134) JTBW(J)	
1629	1131 CONTINUE	
1630	SU(IJ)=SU(IJ)+AW(IJ)*EDT(IJ-NJ)	
1631	BP(IJ)=BP(IJ)+AW(IJ)	
1632	GO TO 1135	
1633	1132 EDT(IJ-NJ)=EDT(IJ)	

Oct 12 1996 16:38	mskemod_2d	Page 24
1634	GO TO 1135	
1635	1133 CONTINUE	
1636	IJ=J	
1637	IPJ=IJ+1	
1638	IMJ=IJ-1	
1639	FYN1=FY(IJ)	
1640	FYN2=FY(IMJ)	
1641	FYS1=1.-FYN1	
1642	FYS2=1.-FYN2	
1643	DYB=YY(IJ)-YY(IMJ)	
1644	DXB=XX(IJ)-XX(IMJ)	
1645	DXBP=QTR*(XX(IJ+NJ)-XX(IJ)+XX(IJ+NJ)-XX(IMJ))	
1646	DYBP=QTR*(YY(IJ+NJ)-YY(IJ)+YY(IJ+NJ)-YY(IMJ))	
1647	FAC=(DXB*DXBP+DYB*DYP)/(DXB**2+DYB**2+SMALL)	
1648	DEL=EDT(IJ)*(FYS1-FYN2)+EDT(IPJ)*FYN1-EDT(IMJ)*FYS2	
1649	EDT(IJ)=EDT(IJ+NJ)-DEL*FAC	
1650	IJ=IJ+NJ	
1651	GO TO 1135	
1652	1134 CONTINUE	
1653	SU(IJ)=GREAT*ED(IJ)	
1654	BP(IJ)=GREAT	
1655	1135 CONTINUE	
1656	AW(IJ)=0.0	
1657	1130 CONTINUE	
1658	C-----EAST BOUNDARY	
1659	DO 1140 J=2,NJM	
1660	IJ=IMNJ(NIM)+J	
1661	GO TO (1141,1142,1143,1144) JTBW(J)	
1662	1141 CONTINUE	
1663	SU(IJ)=SU(IJ)+AE(IJ)*EDT(IJ+NJ)	
1664	BP(IJ)=BP(IJ)+AE(IJ)	
1665	GO TO 1145	
1666	1142 EDT(IJ+NJ)=EDT(IJ)	
1667	GO TO 1145	
1668	1143 CONTINUE	
1669	IJ=IJ+NJ	
1670	IPJ=IJ+1	
1671	IMJ=IJ-1	
1672	FYN1=FY(IJ)	
1673	FYN2=FY(IMJ)	
1674	FYS1=1.-FYN1	
1675	FYS2=1.-FYN2	
1676	DXB=XX(IJ)-XX(IMJ)	
1677	DYB=YY(IJ)-YY(IMJ)	
1678	DXBP=QTR*(XX(IJ-NJ)-XX(IJ)+XX(IJ-NJ)-XX(IMJ))	
1679	DYBP=QTR*(YY(IJ-NJ)-YY(IJ)+YY(IJ-NJ)-YY(IMJ))	
1680	FAC=(DXB*DXBP+DYB*DYP)/(DXB**2+DYB**2+SMALL)	
1681	DEL=EDT(IJ)*(FYS1-FYN2)+EDT(IPJ)*FYN1-EDT(IMJ)*FYS2	
1682	EDT(IJ)=EDT(IJ-NJ)-DEL*FAC	
1683	IJ=IJ-NJ	
1684	GO TO 1145	
1685	1144 CONTINUE	
1686	SU(IJ)=GREAT*ED(IJ)	
1687	BP(IJ)=GREAT	
1688	1145 CONTINUE	
1689	AE(IJ)=0.0	
1690	1140 CONTINUE	
1691	C	
1692	END	
1693	C	
1694	C--- SUBROUTINE 'MODIFY' TO MODIFY SHEAR STRESSES IN	
1695	C--- NEAR WALL GRID POINT	
1696	C-----	
1697	C--- SUBROUTINE MODIFY (SUR, BPR)	
1698	C-----	
1699	C	
1700	INCLUDE 'mskemod.h'	
1701	DIMENSION SUR(NXNY), BPR(NXNY)	
1702	C	
1703	DATA CMU25,CMU75,CAPPA,ELOG/0.5477,0.1643,0.4197,9.0/	
1704	C	

Oct 12 1996 16:38	mskmod_2d	Page 25
1705	DO 10 I=1,NJ	
1706	DO 10 J=1,NJ	
1707	IJ=IMNJ(I)+J	
1708	SU(IJ)=SU(IJ)+SU	
1709	BP(IJ)=BPR(IJ)	
1710	CONTINUE	
1711	C	
1712	C--- CHECK WALL SOUTH BOUNDARY	
1713	C	
1714	DO 600 I=2,NJM	
1715	IJ=IMNJ(I)+2	
1716	IF (ITBS(I).EQ.4) THEN	
1717	LB=IJ	
1718	LM=IJ-1	
1719	TEPR=SQRT(TE(IJ)+TET(IJ))	
1720	DELN=DNS(I)	
1721	DXB=XX(IJ-1)-XX(IJ-NJ-1)	
1722	DYB=YY(IJ-1)-YY(IJ-NJ-1)	
1723	RB=HAF*(R(IJ-1)+R(IJ-NJ-1))	
1724	DENS=DEN(IJ)	
1725	CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,	
1726	& TAU,SUU,SUP,SUV,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)	
1727	SU(IJ)=SU(IJ)+SUU	
1728	BP(IJ)=BP(IJ)+SUP	
1729	SUVS(I)=SVP	
1730	SPVS(I)=SWU	
1731	SPWS(I)=SWP	
1732	GENTS(I)=GENTE	
1733	AS(IJ)=0.0	
1734	ENDIF	
1735	C	
1736	C--- CHECK NORTH WALL-BOUNDARY	
1737	C	
1738	IJ=IMNJ(I)+NJM	
1739	IF (ITBN(I).EQ.4) THEN	
1740	LB=IJ	
1741	LM=IJ+1	
1742	TEPR=SQRT(TE(IJ)+TET(IJ))	
1743	DELN=DNM(I)	
1744	DXB=XX(IJ)-XX(IJ-NJ)	
1745	DYB=YY(IJ)-YY(IJ-NJ)	
1746	RB=HAF*(R(IJ)+R(IJ-NJ))	
1747	DENS=DEN(IJ)	
1748	CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,	
1749	& TAU,SUU,SUP,SUV,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)	
1750	SU(IJ)=SU(IJ)+SUU	
1751	BP(IJ)=BP(IJ)+SUP	
1752	SUVN(I)=SVP	
1753	SPVN(I)=SWU	
1754	SPWN(I)=SWP	
1755	GENTN(I)=GENTE	
1756	AN(IJ)=0.0	
1757	ENDIF	
1758	C	
1759	C	
1760	CONTINUE	
1761	600	
1762	C	
1763	C--- CHECK WALL WEST-BOUNDARY	
1764	C	
1765	DO 620 J=2,NJM	
1766	IJ=IMNJ(2)+J	
1767	IF (ITBW(J).EQ.4) THEN	
1768	LB=IJ	
1769	LM=IJ-NJ	
1770	TEPR=SQRT(TE(IJ)+TET(IJ))	
1771	DELN=DNW(J)	
1772	DXB=XX(IJ-NJ)-XX(IJ-NJ-1)	
1773	DYB=YY(IJ-NJ)-YY(IJ-NJ-1)	
1774	RB=HAF*(R(IJ-NJ)+R(IJ-NJ-1))	
1775	DENS=DEN(IJ)	

Oct 12 1996 16:38	mskmod_2d	Page 26
1776	CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,	
1777	& TAU,SUU,SUP,SUV,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)	
1778	SU(IJ)=SU(IJ)+SUU	
1779	BP(IJ)=BP(IJ)+SUP	
1780	SUVW(J)=SVP	
1781	SPWV(J)=SWP	
1782	SPWW(J)=SWP	
1783	SUWW(J)=SWU	
1784	GENTW(J)=GENTE	
1785	AW(IJ)=0.0	
1786	ENDIF	
1787	C	
1788	C--- CHECK WALL EAST-BOUNDARY	
1789	C	
1790	IJ=IMNJ(NIM)+J	
1791	IF (JTBE(J).EQ.4) THEN	
1792	LB=IJ	
1793	LM=IJ+NJ	
1794	TEPR=SQRT(TE(IJ)+TET(IJ))	
1795	DELN=DNE(J)	
1796	DXB=XX(IJ)-XX(IJ-1)	
1797	DYB=YY(IJ)-YY(IJ-1)	
1798	RB=HAF*(R(IJ)+R(IJ-1))	
1799	DENS=DEN(IJ)	
1800	CALL WALLFN (LB,LW,VISCOS,DENS,DXB,DYB,CMU25,ELOG,CAPPA,	
1801	& TAU,SUU,SUP,SUV,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)	
1802	SU(IJ)=SU(IJ)+SUU	
1803	BP(IJ)=BP(IJ)+SUP	
1804	SUVE(J)=SVP	
1805	SPVE(J)=SWP	
1806	SPWE(J)=SWP	
1807	SUWE(J)=SWU	
1808	GENTER(J)=GENTE	
1809	AE(IJ)=0.0	
1810	ENDIF	
1811	C	
1812	620	
1813	C	
1814	RETURN	
1815	END	
1816	C	
1817	C	
1818	C	
1819	C	
1820	C--- SUBROUTINE 'WALLFN' TO SET WALL FUNCTIONS	
1821	C	
1822	SUBROUTINE WALLFN (LB,LW,VISC,DENS,DXB,DYB,CMU25,ELOG,CAPPA,	
1823	& TAU,SUU,SUP,SUV,SVP,SWU,SWP,GENTE,DELN,TEPR,RB)	
1824	C	
1825	INCLUDE 'mskmod.h'	
1826	C	
1827	UP=U(LB)	
1828	VP=V(LB)	
1829	WP=W(LB)	
1830	UWALL=U(LW)	
1831	VWALL=V(LW)	
1832	WWALL=W(LW)	
1833	ARM=SQRT(DXB**2+DYB**2)	
1834	DXB=DXB/ARM	
1835	DYB=DYB/ARM	
1836	CONST=DENS*CMU25*TEPR	
1837	YPLS=DELN*CONST/VISC	
1838	TCOEFF=VISC/DELN	
1839	IF (LAY2) GOTO 10	
1840	IF (YPLS.LE.11.63) GO TO 10	
1841	UPLUS=LOG(ELOG*YPLS)/CAPPA	
1842	TCOEFF=CONST/UPLUS	
1843	10 CONTINUE	
1844	VPINT=UP*DXB+VP*DYB	
1845	VPX=VPINT*DXB	
1846	VPY=VPINT*DYB	

```

1847 TAU=-TCOEF*(VPX*DXB+VPY*DVB)
1848 ARW=ARW*RB
1849 SUP=TCOEF*ARW*DXB**2
1850 SVP=TCOEF*ARW*DVB**2
1851 SWP=TCOEF*ARW
1852 CON=TCOEF*ARW*DXB*DVB
1853 SUU=-CON*VP+UWALL*TCOEF*ARW*DXB**2
1854 SVU=-CON*UP+WALL*TCOEF*ARW*DVB**2
1855 SWU= TCOEF*ARW*WALL*RB
1856 C
1857 C FOR MOVING WALL
1858 C
1859 VPINT=VPINT*WP
1860 VPINT=ABS(VPINT-SORT(UWALL*UWALL+VWALL*VWALL*WALL))
1861 C
1862 GENTE=TCOEF*CONST*ABS(VPINT)/(CAPPA*DENS*DELN)
1863 C
1864 RETURN
1865 END
1866 C
1867 C-----
1868 C
1869 C-- mskemod.h
1870 C
1871 C COMMON BLOCK
1872 C
1873 C PARAMETER (NX=75)
1874 C PARAMETER (NY=50)
1875 C PARAMETER (NXNY=NK*NY)
1876 C COMMON/KEA1/URFKP,URFKT,URFED,URFV,URFVIS,NI,NJ,NIM,
1877 C & NJM,NINJ,IMNJ(NK),IDIR,ITER,ITBS(NK),ITEN(NK),JTEW(NY),
1878 C & JTB(NY),TEST,AKSI,G
1879 C COMMON/KEA2/XX(NXNY),YY(NXNY),FX(NXNY),FY(NXNY),ARE(NXNY),
1880 C & VOL(NXNY),DMS(NX),DNN(NK),DNN(NY),DNE(NY),R(NXNY)
1881 C COMMON/KEA3/AE(NXNY),AW(NXNY),AN(NXNY),AS(NXNY),AP(NXNY),
1882 C & SU(NXNY)
1883 C COMMON/KEA4/APU(NXNY),APV(NXNY),BE(NXNY),BW(NXNY),BN(NXNY),
1884 C & BS(NXNY),BP(NXNY),RES(NXNY),F1(NXNY),F2(NXNY)
1885 C COMMON/KEA5/U(NXNY),V(NXNY),W(NXNY),P(NXNY),GEN(NXNY),
1886 C & TE(NXNY),TET(NXNY),ED(NXNY),EDT(NXNY),DEN(NXNY),VIS(NXNY),
1887 C & RW(NXNY)
1888 C COMMON/KEA6/RESOR(11),PRTKP,PRTKT,PRTED,PRTEDT,
1889 C & SORMAX,VISCOS,ALFA,GREAT,SMALL,CMU,SRSW(NY),
1890 C & UNW(NY),VNW(NY),PHINW(NY),HAF,QTR,DENSIT,TENOM
1891 C COMMON/KEA7/NSWP(4),SOR(4)
1892 C COMMON/KEA8/ J2LS,J2LN,J2LE,I2LW,LAY2,
1893 C & CPl,CP2,CP3,CT1,CT2,CT3
1894 C COMMON/KEUVMOD/ SUVS(NX),SPVS(NX),SUVN(NX),SPVN(NX),
1895 C & SUVW(NY),SPVW(NY),SUVE(NY),SPVE(NY),
1896 C & SPWN(NX),SPWS(NX),SPWE(NY),SPWW(NY),
1897 C & SUWN(NX),SUWS(NX),SUWE(NY),SUWW(NY)
1898 C COMMON /KEGENER/ GENTS(NX),GENTN(NX),GENTW(NY),GENTEE(NY)
1899 C LOGICAL TEST,AKSI,LAY2

```


CHAPTER 4

2D/Axisymmetric Algebraic Stress Turbulence Model

Table of Contents

	page
4.1 Introduction	49
4.2 Theory and Model Equations	49
4.3 Module Evaluation	52
References	53
Figures	
Appendix C	54

4.1 Introduction

In this section a description is given of the two-dimensional/Axisymmetric Algebraic Stress turbulence Model (ASM) based on the work of Rodi [1]. The model is coded as a self contained computer program to compute turbulent flow quantities when interfaced with a CFD solver. Detailed description of the module structure, variables used and how to interface the module with CFD flow solvers are given in Appendix C.

The module uses as input the mean flow properties, as computed by conventional CFD solvers, and calculates the Reynolds stresses, turbulent kinetic energy and the energy dissipation. It is structured to be self-contained and compatible with many CFD codes. It has been tested as a separate unit at Rocketdyne using the finite-volume REACT code [2]. The module has also been tested independently at the University of Alabama at Huntsville (UAH) using own code MAST.

The module computes turbulent flow quantities in two-dimensional planar or axisymmetric geometry with or without swirl. The standard wall functions and the two-layer model of Chen and Patel [3] are used for the near wall treatment.

4.2 Theory and Model Equations

The Algebraic Stress (ASM) module is based on the work of Rodi [1]. The idea is to simplify or truncate the Reynolds stress equation by approximating the convective and diffusive transport of the Reynolds stresses $\overline{u_i u_j}$ in terms of the corresponding transport of turbulent energy. This allows the transport equation for the stresses to be expressed as a set of algebraic formulae containing the turbulence energy and its rate of dissipation as unknowns in the form:

$$\overline{u_i u_j} = \frac{k}{(P-\epsilon)} \left[P_{ij} - \frac{2}{3} \delta_{ij} \epsilon + \Phi_{ij} \right]$$

where P_{ij} = Production and $P = \frac{1}{2} P_{kk}$ and

Φ_{ij} = pressure-strain redistribution

$$\Phi_{ij} = \Phi_{ij,1} + \Phi_{ij,2} + \Phi_{ij,1w} + \Phi_{ij,2w}$$

Rotta's linear return-to-isotropy concept for the non-linear part of

$$\Phi_{ij,1} = -C_1 \frac{\varepsilon}{k} (\overline{u_i u_j} - \frac{2}{3} k \delta_{ij})$$

is used and the "isotropization of production" concept for the linear "rapid" part of

$$\Phi_{ij,2} = -C_2 (P_{ij} - \frac{2}{3} P \delta_{ij})$$

is used. Gibson and Launder [4] concept for the wall reflection terms is used as

$$\Phi_{ij,1w} = C_{1w} \rho \frac{\varepsilon}{k} (\overline{u_k u_m} n_k n_m \delta_{ij} - \frac{3}{2} \overline{u_k u_i} n_k n_j - \frac{3}{2} \overline{u_k u_j} n_k n_i) f$$

$$\Phi_{ij,2w} = C_{2w} (\Phi_{km,2} n_k n_m \delta_{ij} - \frac{3}{2} \Phi_{ik,2} n_k n_j - \frac{3}{2} \Phi_{jk,2} n_k n_i) f$$

where (n_i) is the wall-normal unit vector in the i -direction. The wall-distance function (f) represents the ratio of the turbulence length scale ($L_\varepsilon = \frac{k^{3/2}}{\varepsilon}$) and the wall distance and is given as

$$f = (\frac{C_m^{0.75} k^{1.5}}{K \varepsilon}) \frac{1}{\Delta n}$$

with Δn being the wall-normal distance.

The set of algebraic stress equations can be arranged in the form

$$A_{ij} \overline{u^2} + B_{ij} \overline{v^2} + C_{ij} \overline{w^2} + D_{ij} \overline{uv} + E_{ij} \overline{vw} + F_{ij} \overline{uw} = G_{ij}$$

where A_{ij} , B_{ij} , C_{ij} , D_{ij} , E_{ij} , F_{ij} , and G_{ij} are functions of the mean and turbulent flow variables.

The above equation can be solved iteratively in the main flow solver. However, the algebraic system of equations is stiff and convergence difficulties are encountered when solved iteratively. Therefore, the set of equations was cast in the general matrix form $\underline{\mathbf{A}} \underline{\mathbf{T}} = \underline{\mathbf{B}}$, where

$$\begin{aligned}
\underline{\mathbf{A}} = & \begin{array}{cccccc}
\frac{3\varepsilon}{2\lambda\kappa} + 2\frac{\partial U}{\partial x} & -\frac{\partial V}{\partial y} & -\frac{V}{r} & 2\frac{\partial U}{\partial y} - \frac{\partial V}{\partial x} & -\frac{\partial W}{\partial y} + \frac{W}{r} & -\frac{\partial W}{\partial x} \\
-\frac{\partial U}{\partial x} & \frac{3\varepsilon}{2\lambda\kappa} + 2\frac{\partial V}{\partial y} & -\frac{V}{r} & 2\frac{\partial V}{\partial x} - \frac{\partial U}{\partial y} & -(\frac{\partial W}{\partial y} + 2\frac{W}{r}) & -\frac{\partial W}{\partial x} \\
-\frac{\partial U}{\partial x} & -\frac{\partial V}{\partial y} & \frac{3\varepsilon}{2\lambda\kappa} + 2\frac{V}{r} & -(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}) & 2\frac{\partial W}{\partial y} + \frac{W}{r} & 2\frac{\partial W}{\partial x} \\
\frac{\partial V}{\partial x} & \frac{\partial U}{\partial y} & 0 & \frac{\varepsilon}{\lambda\kappa} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} & 0 & -\frac{W}{r} \\
0 & \frac{\partial W}{\partial y} & -\frac{W}{r} & \frac{\partial W}{\partial x} & \frac{\varepsilon}{\lambda\kappa} + \frac{\partial V}{\partial y} + \frac{V}{r} & \frac{\partial V}{\partial x} \\
\frac{\partial W}{\partial x} & 0 & 0 & \frac{\partial W}{\partial y} & \frac{\partial U}{\partial y} & \frac{\varepsilon}{\lambda\kappa} + \frac{\partial U}{\partial x} + \frac{V}{r}
\end{array}
\end{aligned}$$

$$\underline{\mathbf{T}} = [\rho \overline{uu}, \rho \overline{vv}, \rho \overline{ww}, \rho \overline{uv}, \rho \overline{vw}, \rho \overline{uw}]^T$$

$$\begin{aligned}
\underline{\mathbf{B}} = & \begin{array}{l}
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)} (\Phi_{11,1w} + \Phi_{11,2w}) \\
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)} (\Phi_{22,1w} + \Phi_{22,2w}) \\
\frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)} (\Phi_{33,1w} + \Phi_{33,2w}) \\
\frac{1}{(1-C_2)} (\Phi_{12,1w} + \Phi_{12,2w}) \\
\frac{1}{(1-C_2)} (\Phi_{23,1w} + \Phi_{23,2w}) \\
\frac{1}{(1-C_2)} (\Phi_{13,1w} + \Phi_{13,2w})
\end{array}
\end{aligned}$$

$$\text{where } \lambda = \frac{1-C_2}{C_1-1 + \frac{P}{\rho\varepsilon}}$$

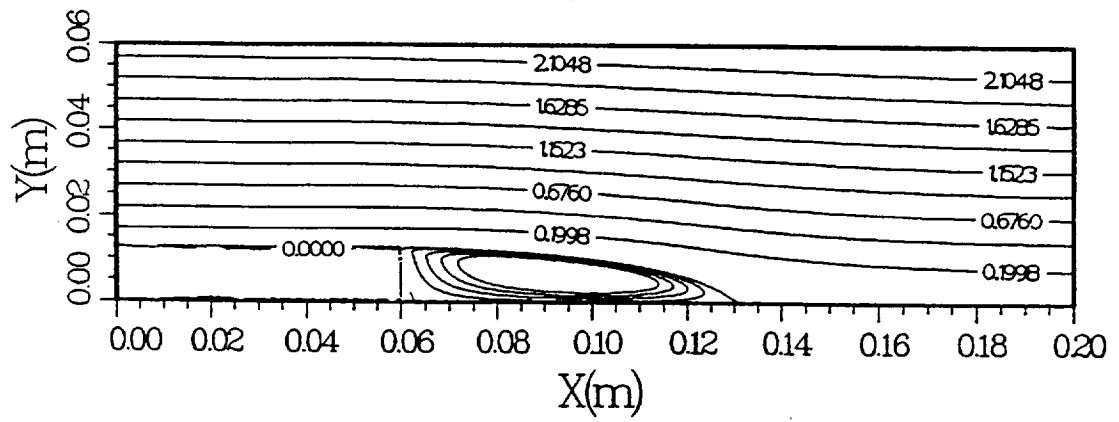
The matrix was inverted at each iteration step to obtain a converged solution. The wall function and a two-layer model were built in the module to model the near-wall region.

4.3 Module Evaluation

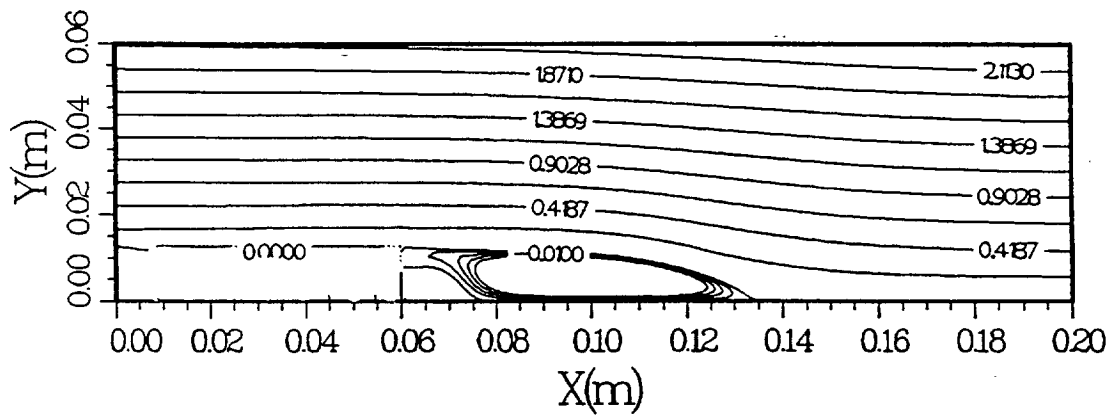
The ASM module was evaluated by comparison with experimental data of Driver and Seegmiller [5] for the backward facing step and the data of Roback and Johnson [6]. The effect of the wall reflection term is also studied with both wall function and two-layer near wall models. Figures 1a and 1b show the stream-function contours for a backward facing step flow using the wall function and the two-layer near wall models with reattachment length of $5.59H$ and $5.83H$ respectively (H is the step height). Figures 2a and 2b show the stream-function contours for the Roback & Johnson confined swirling jet flow using the wall function near wall model, where (a) includes the wall-reflection term in the pressure-strain redistribution term and (b) without the wall reflection term. Figure 3 shows a comparison of the axial velocity along the centerline with and without wall reflection term. The comparisons of the predicted mean axial velocity, mean tangential velocity, turbulent intensities $\overline{uu}^{1/2}$, $\overline{vv}^{1/2}$, $\overline{ww}^{1/2}$, and the Reynolds stress $\overline{uw}^{1/2}$ using the ASM model as compared with the single and multi-scale $k-\epsilon$ models are presented in figures 4 to 9 respectively. The figures in general show that the ASM model used here when combined with the wall function near wall treatment predicts better comparisons without using the wall reflection terms. This may be explained by the fact that the wall reflection terms -whose purpose is to damp normal turbulent intensity normal to the wall as the wall is reached- are not effective when using wall functions near the wall. Similar conclusions were also obtained by the UAH group when testing the ASM module using their code (MAST). Also, in the ASM model, a set of algebraic equations for the Reynolds stresses are solved and there is no boundary conditions are needed for the stresses. This is not the same in the full Reynolds stress model (RSM) where a set of nonlinear differential equations are solved and boundary conditions for the stresses are required. More on this will be discussed in detail in the next RSM module. Also, more details will be given on the tensorial incorporation of the wall reflection terms since they are tied to the orientation of the wall through the unit normal vectors.

REFERENCES

- [1] Rodi, W., 'A New Algebraic Stress Relations for Calculating the Reynolds Stresses' Z. Ang. Math. und Mech., vol. 56, pp. 219-, 1976.
- [2] Chon J., Hadid A. and Hamakiotes C. 'REACT-2D Version 2.6 User's Manual' CFD Technology Center, Rocketdyne Division/Rockwell International, 1989.
- [3] Chen H. and Patel V. 'Near-Wall Turbulence Models for Complex Flows Including Separation', AIAA Journal, vol. 26, No. 6, pp. 641-648, 1988.
- [4] Gibson M. and Launder B., 'Ground Effects on Pressure Fluctuations in the Atmosphere boundary Layer', J. Fluid Mech. vol. 86, pt. 3, pp. 491-., 1978.
- [5] Driver, D. and Seegmiller, H. "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow" AIAA J., vol. 23, pp. 163-171, 1985.
- [6] Roback, R. and Johnson, B. "Mass and Momentum Turbulent Transport Experiment with Confined Swirling Co-axial Jets" NASA CR-168252, 1983.
- [7] Daly B. and Harlow F., 'Transport Equation in Turbulence', Physics of Fluids, vol. 13, pp. 2634-., 1970.
- [8] Stone H., 'Iterative Solution of Implicit Approximations of Multi-Dimensional Partial Differential Equations', SIAM J. Num. Anal., vol 5, pp. 530- ,1968.

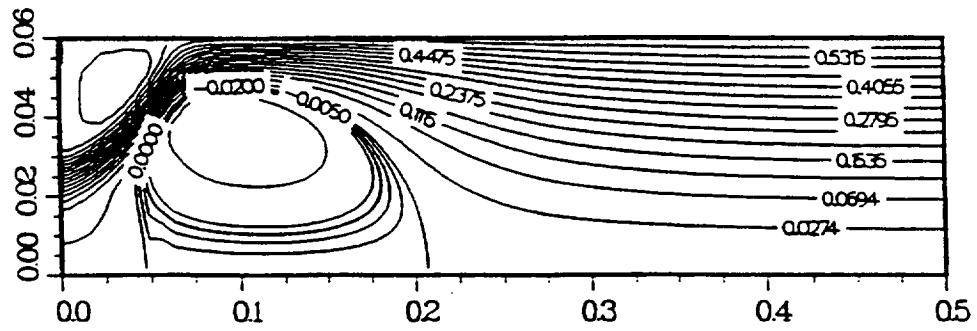


(a)

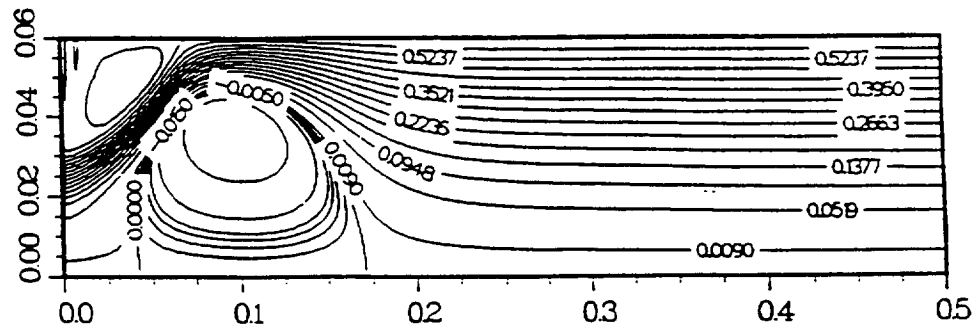


(b)

Figure 1. Stream-function contours of backward facing step flow using the ASM with (a) wall function and (b) two-layer near wall treatment



(a) ASM with Φ_w term



(b) ASM without Φ_w term

Figure 2. Stream-function contours of confined swirling jet flow

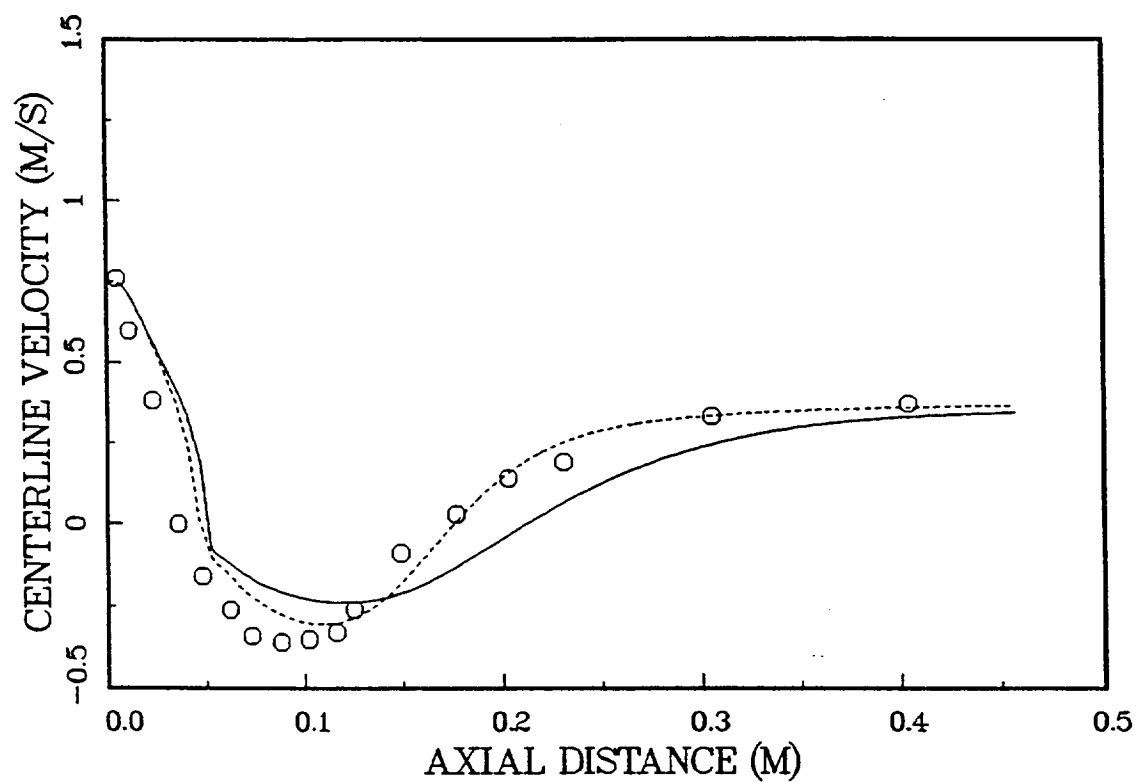
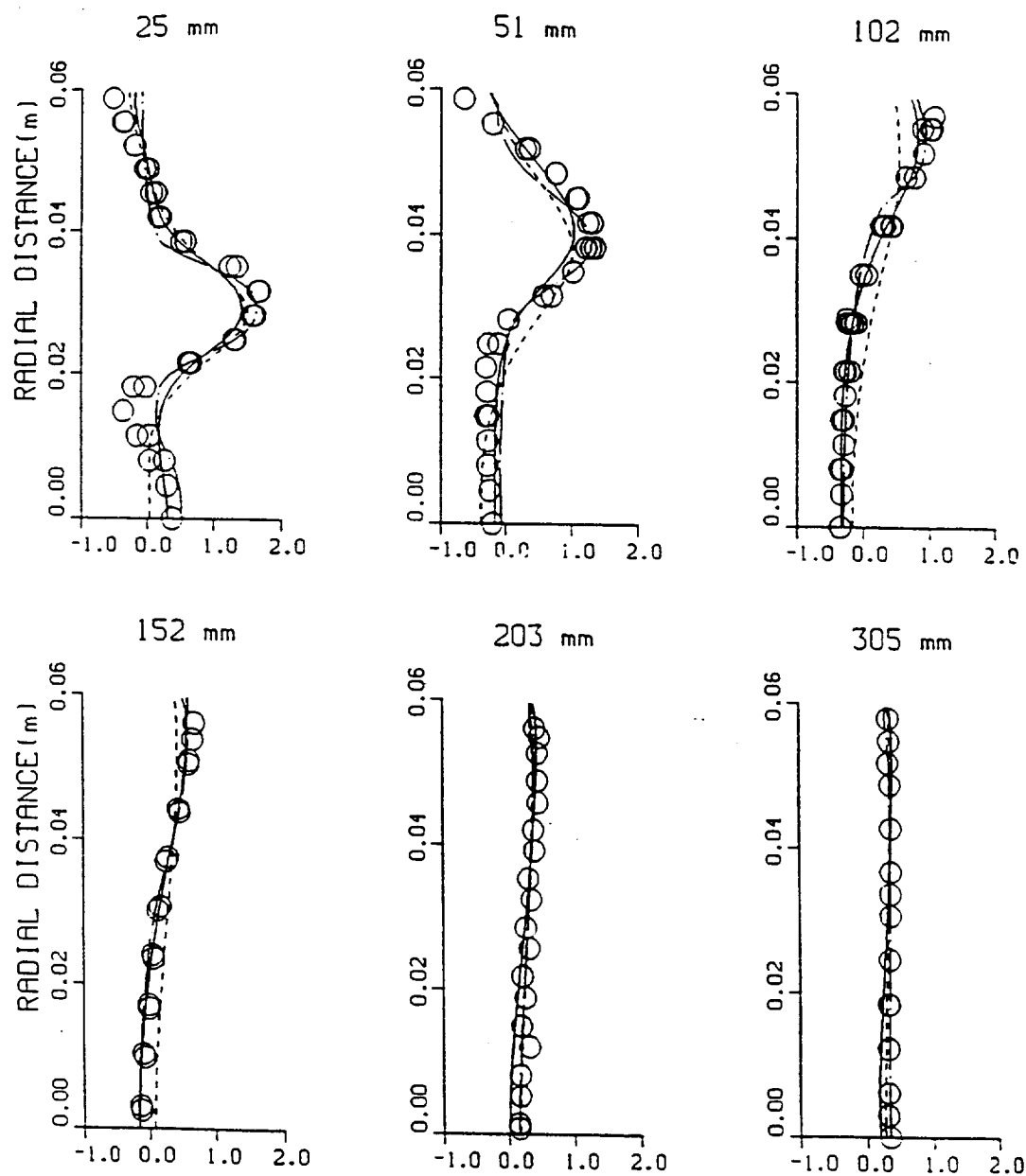


Figure 3. Decay of axial velocity along the centerline in confined swirling jet flow

- o Roback & Johnson
- ASM no Φ_w
- ASm with Φ_w



— M-S model, --- ASM, . . . $k-\epsilon$ model

Figure 4. Radial profiles of mean axial velocity in confined swirling jet flow

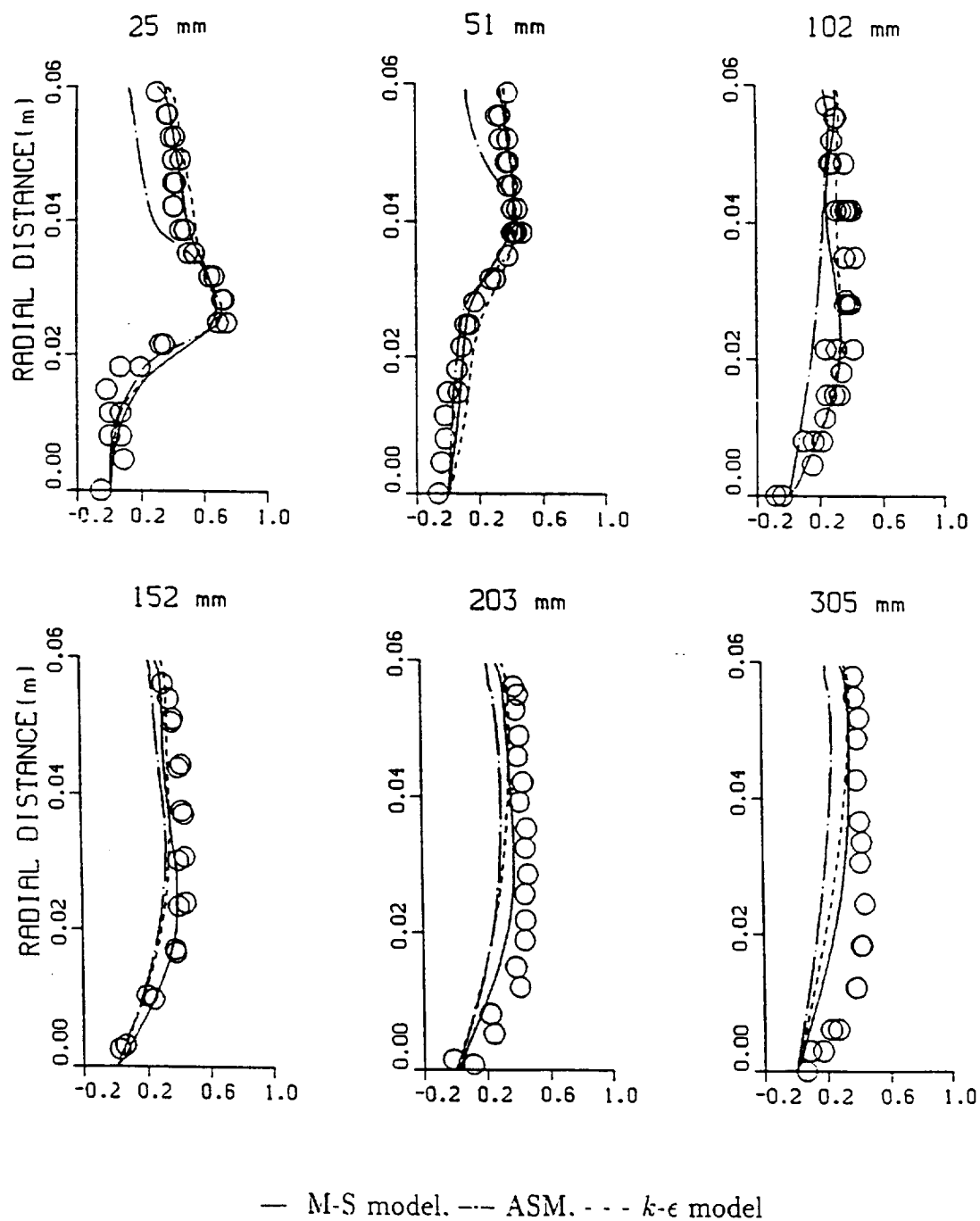


Figure 5. Radial profiles of mean tangential velocity in confined swirling jet flow

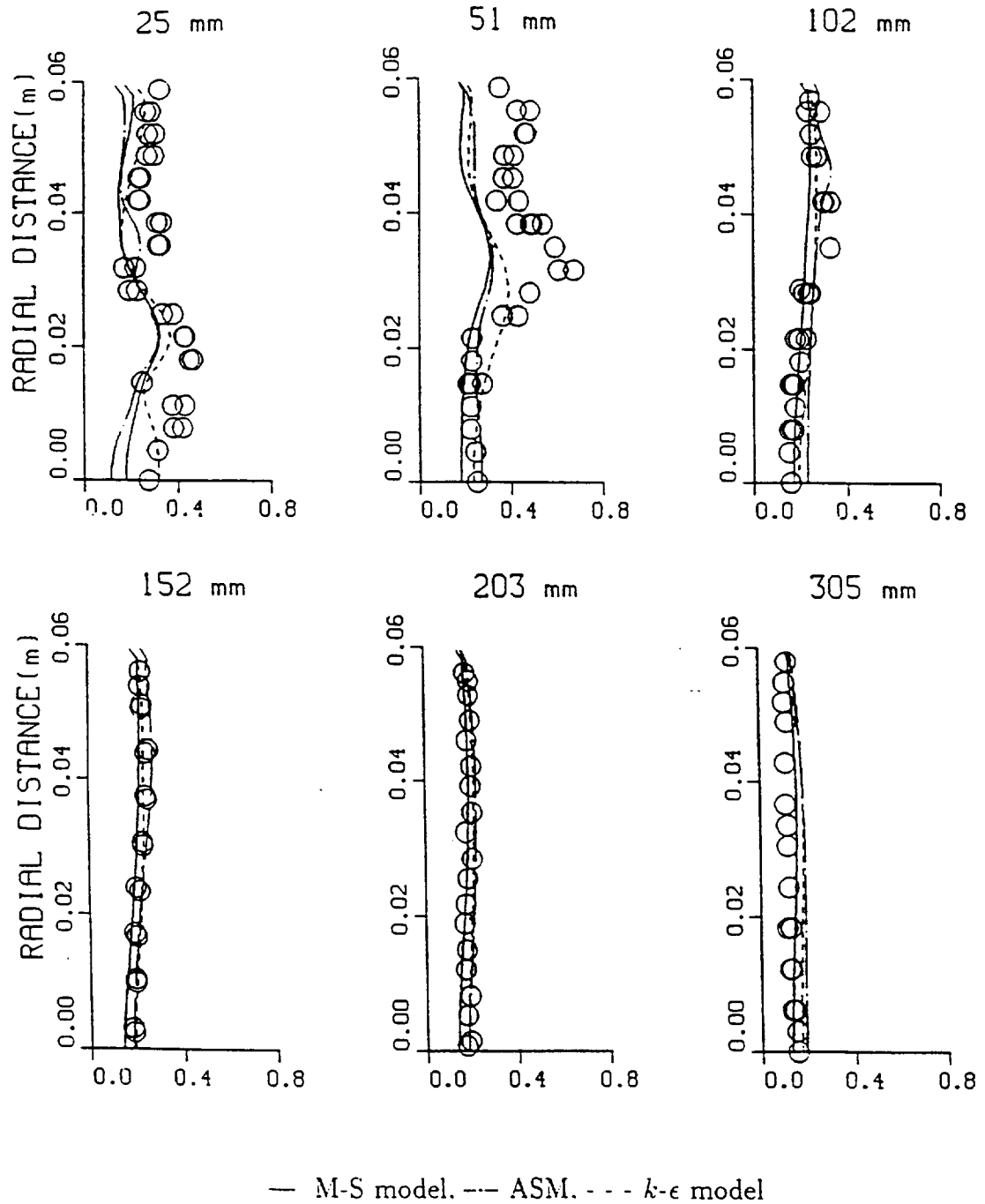


Figure 6. Radial profiles of turbulent intensity $(\overline{uu})^{1/2}$ in confined swirling jet flow

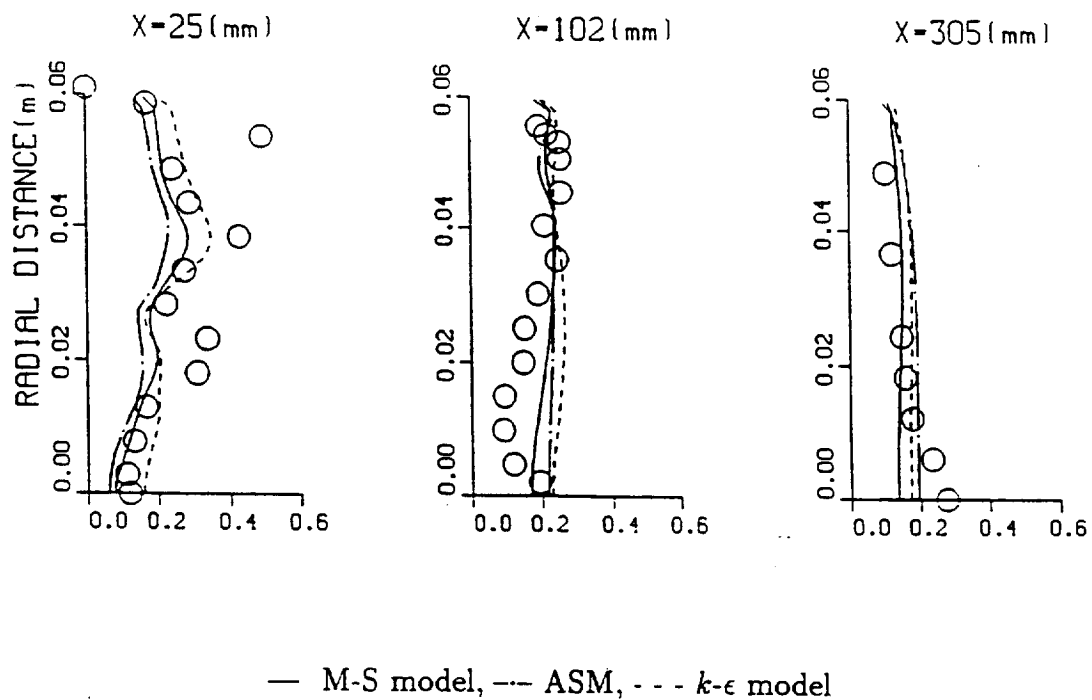


Figure 7. Radial profiles of turbulent intensity $(\overline{v'v'})^{1/2}$ in confined swirling jet flow

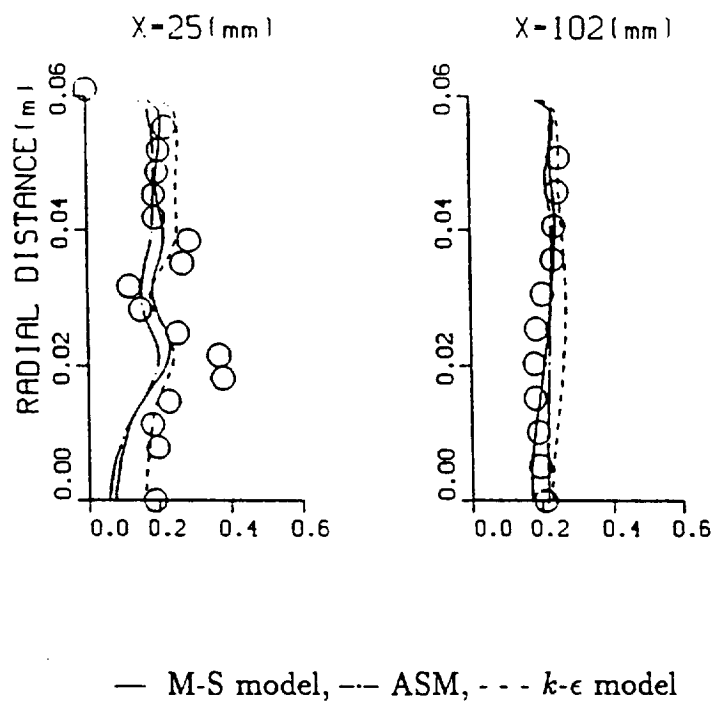


Figure 8. Radial profiles of turbulent intensity $(\overline{w'w'})^{1/2}$ in confined swirling jet flow

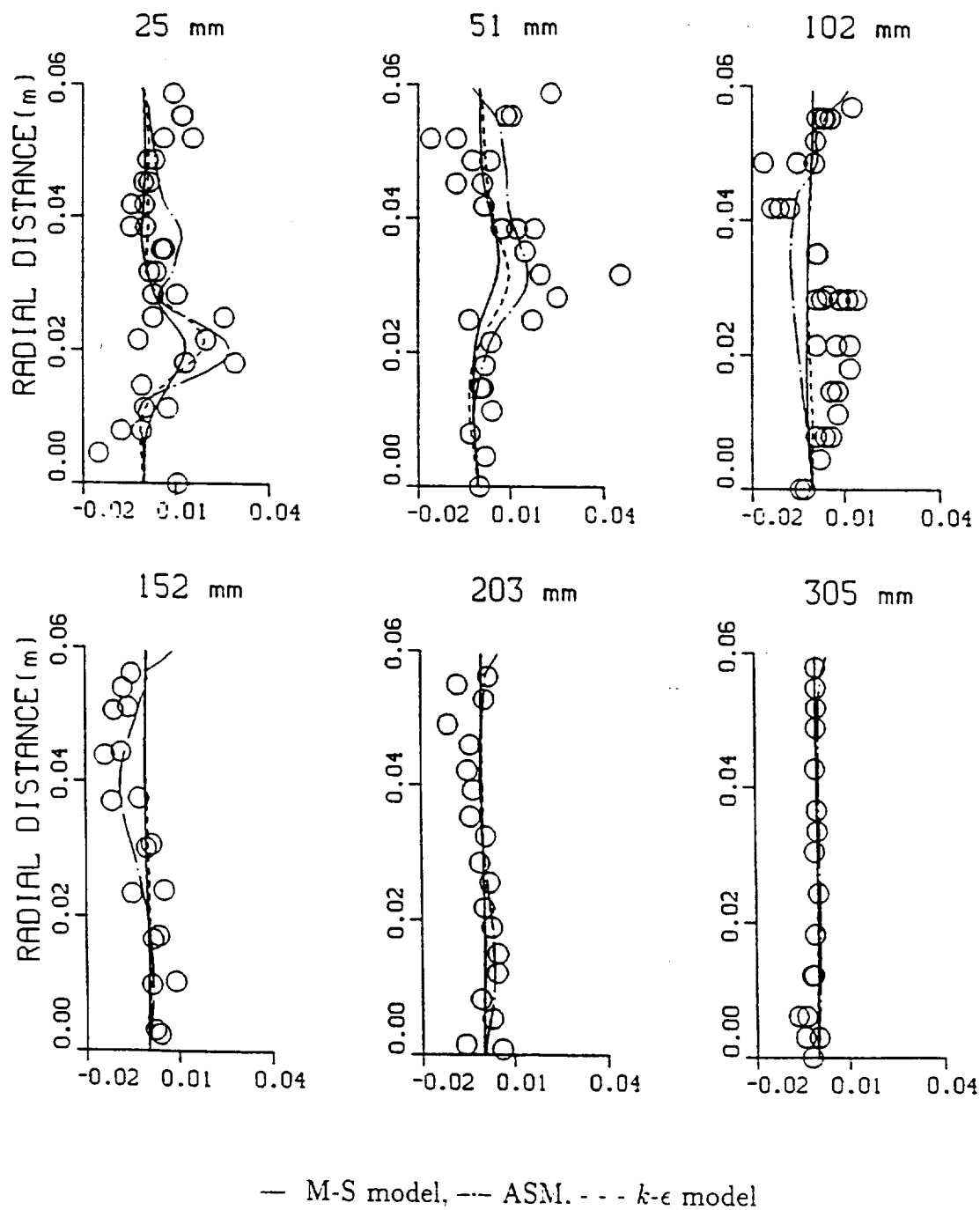


Figure 9. Radial profiles of turbulent shear stress \overline{uw} in confined swirling jet flow

APPENDIX C

2D/Axisymmetric Algebraic Stress Turbulence Module Deck

This module is a FORTRAN source code to solve 2D/Axisymmetric turbulent flow quantities using the algebraic stress model when interfaced with a main flow solver. The module consists of the main routine ASMOD that calls a number of subroutines to perform different functions that will be explained below.

3.1 Subroutine ASMOD

This is basically the main routine that reads through its argument list different variables from the calling flow solver which are described below.

List of Argument Variable Names

X	Grid node locations in the x or ξ -direction, dimensioned to X(NX*NY)
Y	Grid node locations in the y or η -direction, dimensioned to Y(NX*NY)
FX	Interpolation factor in the x or ξ -direction.
FY	Interpolation factor in the y or η -direction.
ARE	Cell areas
VOL	Cell volumes.
R	Radial distance in the axisymmetric geometry or 1. for planar geometry.
DNS	Normal distance of a cell from the south-boundary dimensioned to NX.
DNN	Normal distance of a cell from the north-boundary dimensioned to NX.
DNE	Normal distance of a cell from the east-boundary dimensioned to NY.
DNW	Normal distance of a cell from the west-boundary dimensioned to NY.
U	Axial or x-direction velocity, dimensioned to NX*NY.
V	Radial or y-direction velocity, dimensioned to NX*NY.
W	Tangential or azimuthal velocity, dimensioned to NX*NY.
TE	Turbulent kinetic energy, dimensioned to NX*NY.
ED	Turbulent energy dissipation rate, dimensioned to NX*NY.
DEN	Density (assumed constant for incompressible flows).
F1	Mass flux at cell faces in the x or ξ -direction, dimensioned to NX*NY.
F2	Mass flux at cell faces in the y or η -direction, dimensioned to NX*NY.

VISCOUS	Laminar viscosity.
VIS	Eddy viscosity, dimensioned to NX*NY.
RESOR	Residual error for the k and ϵ -equations solver, dimensioned to 2.
ITBS	Boundary condition flag along the south boundary dimensioned to NX and must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall, e.g., for a wall boundary condition along the south boundary set ITBS to NX*4. Similarly for the other boundaries.
ITBN	Boundary condition flag along the north boundary, dimensioned to NX.
JTBE	Boundary condition flag along the east-boundary dimensioned to NY.
JTBW	Boundary condition flag along the west-boundary dimensioned to NY.
ITER	Iteration number.
FMUU	function used in the two-layer model.
ICAL	= 1 for swirl velocity calculations, 0 otherwise.
AKSI	= 1 for axisymmetric flow, 0 otherwise.
RESTART	= 1 if calculations are restarted from a previous run, 0 otherwise.

ASMOD starts by reading the turbulent flow constants, under-relaxation factors and Prandtl/Schmidt numbers for the k and ϵ equations. These are;

CD1, CD2	constants in the k and ϵ -equations and are usually set to 1.44 and 1.92 respectively.
CMU, ELOG, and CAPPA	constants in the k and ϵ -equations and are usually set to 0.09, 9.8 and 0.42 respectively.
LAY2	set to true (T) for two-layer model and false (F) for wall functions.
GKE	is set to 1 for second-order upwinding of the convective terms in the k and ϵ -equations.
ALFAKE	is the iteration parameter used in the k and ϵ -equation solver.
URFVIS	is the underrelaxation factor of the viscosity near the wall.
SORKE(1) and SORKE (2)	are the degree of accuracy for the k and ϵ -equation solver respectively.
URFKE(1) and URFKE(2)	are the underrelaxation factors for the k and ϵ -equations respectively.
PRTKE(1) and PRTKE(2)	are ratio of Prandtl to Schmidt numbers used in the k and ϵ -equations in the two-layer model near the wall.
C1, C2	are constants in the ASM model.

C1W and C2W	are the two constants in the wall-reflection terms of the pressure-strain redistribution term.
CK and CE	constants in the diffusion term of the k and ε -equations.
WREFON	= 1 if the wall reflection terms of the pressure-strain term are to be included, 0 otherwise.

All dimensions considered are one-dimensional. The position of any node is defined as $IJ = (I, J) = (I-1)*NJ + J$, where NI and NJ are the number of grid nodes in the X and Y -directions respectively. It is assumed that grid related data such as cell areas, volumes and interpolation factors be passed to the module from an external grid generator.

Subroutine WALREF

This subroutine calculates the wall reflection terms in the pressure-strain redistribution term. It calculates the wall unit normal vectors and the normal distance away from the wall. This is needed to resolve the wall tangential and normal velocity components that are needed to obtain the near-wall values of the Reynolds stresses.

Subroutine CALPIJ

This subroutine calculates the production terms of the individual stress components.

Subroutine CALUIUJ

This subroutine calculates the individual stress component from its algebraic equation. It sets the coefficients of the algebraic stress equations which are solved implicitly at each iteration step by inverting a 6x6 matrix.

Subroutine ACALCKE

This subroutine solves the transport equations for the turbulent energy (IPHI=1) and energy dissipation.(IPHI=2). Daly and Harlow [7] gradient stress diffusion form is used in the module instead of the simplified isotropic diffusivity form. The subroutine calls MODPHI subroutine that sets the appropriate boundary conditions for k and ε . The set of algebraic difference equations are then solved using Stone's strongly implicit solver ASOLSIP.

Subroutine ATWOLAY

This subroutine calculates the near wall turbulence using Chen and Patel's [3] two layer model.

Subroutine MODPIJ

This subroutine modifies the production terms near the wall using the near wall region model.

Subroutine MODPHI

This subroutine calculates the near wall boundary conditions for the turbulence energy and the energy dissipation.

Subroutine AMODVIS

This subroutine modifies the eddy viscosity close to a wall using the near wall model chosen.

Subroutine ASOLSIP

This subroutine solves the system of linear algebraic equations for k and ε using Stone's Implicit Procedure [8].

Subroutine AMODIFY

This subroutine is called from the momentum equations solver of the main routine. It updates the flux source terms of the discretized momentum equations due to wall shear stresses and due to the Reynolds stress gradients. The terms SUASM, SVASM and SWASM need to be added to the U, V and W-momentum equations of the main solver. They represent the difference form of the Reynold stress gradients in the momentum equations.

Oct 12 1996 16:36 asmod_2d Page 1

asmod_2d Page 2

[illegible]

```

72 C--- CALCULATE TURBULENT KINETIC ENERGY.
73 C
74 C   IPI=1
75 C   DO 20 IJ=1,NINJ
76 C   PHI(IJ)=TE(IJ)
77 C
78 C   CALL ACALCKE(PHI,X,Y,FX,FY,ARE,VOL,R,DNS,DNN,DNE,DNW,
79 C   & U,V,W,DEN,F1,F2,VIS,VISCOS,TE,ED,
80 C   & ITBS,ITBN,JTBE,JTBW,AKSI,IPI,RESOR)
81 C
82 C   DO 25 IJ=1,NINJ
83 C   TE(IJ)=PHI(IJ)
84 C
85 C   IF (LAY2) CALL ATWOLAY(X,Y,DEN,TE,ED,VISCOS,ITBS,ITBN,JTBE,
86 C   & CALCULATE TURBULENT KINETIC ENERGY DISSIPATION
87 C---
88 C
89 C   IPI=2
90 C   DO 30 IJ=1,NINJ
91 C   PHI(IJ)=ED(IJ)
92 C
93 C   CALL ACALCKE(PHI,X,Y,FX,FY,ARE,VOL,R,DNS,DNN,DNE,DNW,
94 C   & U,V,W,DEN,F1,F2,VIS,VISCOS,TE,ED,
95 C   & ITBS,ITBN,JTBE,JTBW,AKSI,IPI,RESOR)
96 C
97 C   DO 35 IJ=1,NINJ
98 C   ED(IJ)=PHI(IJ)
99 C
100 C   CALL AMODVIS (TE,ED,VIS,DEN,VISCOS,
101 C   & ITBS,ITBN,JTBE,JTBW)
102 C--- CHECK TURBULENT ENERGY CALCULATED (TE) WITH BELOW (TERS)
103 C
104 C   DO 40 IJ=1,NINJ
105 C   FMU(IJ)=FMU(IJ)
106 C   TERS(IJ)=0.5*(U2(IJ)+V2(IJ)+W2(IJ))
107 C   CONTINUE
108 C
109 C   if(mod(iter,10).eq.1) write(*,7) TERS(923)/DEN(923),TE(923)
110 C   format(1x,7(1x,E10.3))
111 C   RETURN
112 C   END
113 C
114 C--- SUBROUTINE WALREF (X,Y,TE,ED,ITBS,ITBN,JTBW,JTBE)
115 C
116 C--- INCLUDE 'gridparam.h'
117 C
118 C   INCLUDE 'asm.h'
119 C
120 C   DIMENSION X(NXNY),Y(NXNY),TE(NXNY),ED(NXNY)
121 C   DIMENSION FN1S(NX),FN1N(NX),FN1E(NY),FN1W(NY)
122 C   DIMENSION FN2S(NX),FN2N(NX),FN2E(NY),FN2W(NY)
123 C   DIMENSION FT1S(NX),FT1N(NX),FT1E(NY),FT1W(NY)
124 C   DIMENSION FT2S(NX),FT2N(NX),FT2E(NY),FT2W(NY)
125 C   DIMENSION ITBS(NX),ITBN(NX),JTBW(NX),JTBE(NY)
126 C--- CALCULATE NORMAL AND TANGENTIAL WALL UNIT VECTORS
127 C
128 C   NI=NIN+1
129 C   NJ=NJM+1
130 C--- ALONG SOUTH & NORTH WALLS---
131 C
132 C   DO 10 I=2,NIM
133 C   IF(ITBS(I).EQ.4) THEN
134 C     IJ=IMNJ(I)+1
135 C     DXB=X(IJ)-Y(IJ-NJ)
136 C     DYB=Y(IJ)-Y(IJ-NJ)
137 C     FHP=SQRT(DXB**2+DYB**2)
138 C     FT1S(I)=DXB/FHP
139 C     FT2S(I)=DYB/FHP
140 C     FN1S(I)=-DXB/FHP
141 C     FN2S(I)=DXB/FHP
142 C   ENDIF

```

Oct 12 1996 16:36	asmod_2d	Page 3
143	IF (ITBN(I).EQ.4) THEN	
144	IJ=IMNJ(I)-NJ	
145	DXB=X(IJ)-X(IJ-NJ)	
146	DYB=Y(IJ)-Y(IJ-NJ)	
147	PHIP=SQRT(DXB**2+DYB**2)	
148	FT1N(I)=DXB/PHIP	
149	FT2N(I)=DYB/PHIP	
150	FN1N(I)=DXB/PHIP	
151	FN2N(I)=DXB/PHIP	
152	ENDIF	
153	CONTINUE	
154	C	
155	C--- ALONG WEST & EAST BOUNDARIES ---	
156	C	
157	DO 20 J=2,NJM	
158	IF (JTBW(J).EQ.4) THEN	
159	IJ=J	
160	DXB=X(IJ)-X(IJ-1)	
161	DYB=Y(IJ)-Y(IJ-1)	
162	PHIP=SQRT(DXB**2+DYB**2)	
163	FT1W(J)=DXB/PHIP	
164	FT2W(J)=DYB/PHIP	
165	FN1W(J)=DXB/PHIP	
166	FN2W(J)=DXB/PHIP	
167	ENDIF	
168	IF (JTBE(J).EQ.4) THEN	
169	IJ=IMNJ(NIM)+J	
170	DXB=X(IJ)-X(IJ-1)	
171	DYB=Y(IJ)-Y(IJ-1)	
172	PHIP=SQRT(DXB**2+DYB**2)	
173	FT1E(J)=DXB/PHIP	
174	FT2E(J)=DYB/PHIP	
175	FN1E(J)=DXB/PHIP	
176	FN2E(J)=DXB/PHIP	
177	ENDIF	
178	CONTINUE	
179	C	
180	CMU25=SQRT(SQRT(CMU))	
181	CMU75=CMU25**3	
182	FCON=CMU75/CAPPA	
183	C	
184	DO 80 I=2,NIM	
185	DO 80 J=2,NJM	
186	IJ=IMNJ(I)+J	
187	RDISNS=0.0	
188	RDISNW=0.0	
189	RDISNE=0.0	
190	RDISNW=0.0	
191	TE(IJ)=ABS(TE(IJ))	
192	CORF=FCON*TE(IJ)**1.5/(ED(IJ)+SMALL)	
193	C--- START WITH SOUTH BOUNDARY	
194	IF (ITBS(I).EQ.4) THEN	
195	IJW=IMNJ(I)+1	
196	IJW=IJW-NJ	
197	DXB=X(IJW)-X(IMJW)	
198	DYB=Y(IJW)-Y(IMJW)	
199	XB=HAF*(X(IJW)+X(IMJW))	
200	YB=HAF*(Y(IJW)+Y(IMJW))	
201	YBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
202	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
203	DXBP=XBP-XB	
204	DYBP=YBP-YB	
205	DISNS=DELTA(DXB,DYB,DXBP,DYBP)	
206	DISNS=1.0/(DISNS+SMALL)	
207	ENDIF	
208	C---CHECK NORTH BOUNDARY	
209	IF (ITBN(I).EQ.4) THEN	
210	IJW=IMNJ(I)+NJM	
211	IJW=IJW-NJ	
212	DXB=X(IJW)-X(IMJW)	
213	DYB=Y(IJW)-Y(IMJW)	

Oct 12 1996 16:36	asmod_2d	Page 4
214	XB=HAF*(X(IJW)+X(IMJW))	
215	YB=HAF*(Y(IJW)+Y(IMJW))	
216	XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
217	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
218	DXBP=XBP-XB	
219	DYBP=YBP-YB	
220	DISNN=DELTA(DXB,DYB,DXBP,DYBP)	
221	RDISNN=1.0/(DISNN+SMALL)	
222	ENDIF	
223	C---ALONG THE WEST BOUNDARY	
224	IF (JTBW(J).EQ.4) THEN	
225	IJW=J	
226	IMJW=IJW-1	
227	DXB=X(IJW)-X(IMJW)	
228	DYB=Y(IJW)-Y(IMJW)	
229	XB=HAF*(X(IJW)+X(IMJW))	
230	YB=HAF*(Y(IJW)+Y(IMJW))	
231	XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
232	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
233	DXBP=XBP-XB	
234	DYBP=YBP-YB	
235	DISNW=DELTA(DXB,DYB,DXBP,DYBP)	
236	RDISNW=1.0/(DISNW+SMALL)	
237	ENDIF	
238	C---CHECK EAST BOUNDARY	
239	IF (JTBE(J).EQ.4) THEN	
240	IJW=IMNJ(NIM)+J	
241	IMJW=IJW-1	
242	DXB=X(IJW)-X(IMJW)	
243	DYB=Y(IJW)-Y(IMJW)	
244	XB=HAF*(X(IJW)+X(IMJW))	
245	YB=HAF*(Y(IJW)+Y(IMJW))	
246	XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
247	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
248	DXBP=XBP-XB	
249	DYBP=YBP-YB	
250	DISNE=DELTA(DXB,DYB,DXBP,DYBP)	
251	RDISNE=1.0/(DISNE+SMALL)	
252	ENDIF	
253	C	
254	FUNX(IJ)=COEF*(RDISNS*FN1S(I)**2+RDISNN*FN1N(I)**2+	
255	& RDISNE*FN1E(J)**2+RDISNW*FN1W(J)**2)	
256	FUNY(IJ)=COEF*(RDISNS*FN2S(I)**2+RDISNN*FN2N(I)**2+	
257	& RDISNE*FN2E(J)**2+RDISNW*FN2W(J)**2)	
258	FUNXY(IJ)=COEF*(RDISNS*FN1S(I)*FN2S(I)+	
259	& RDISNN*FN1N(I)*FN2N(I)+RDISNE*FN1E(J)*FN2E(J)+	
260	& RDISNW*FN1W(J)*FN2W(J))	
261	C	
262	C	
263	RETURN	
264	END	
265	C	
266	C	
267	C	
268	SUBROUTINE CALPIJ (X,Y,FX,FY,ARE,VOL,R,ICAL,AKSI,U,V,W,	
269	& ITBS,ITBN,JTBE,JTBW,ITER)	
270	C	
271	INCLUDE 'gridparam.h'	
272	INCLUDE 'asm.h'	
273	C	
274	DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY)	
275	DIMENSION ARE(NXNY),VOL(NXNY),R(NXNY)	
276	DIMENSION U(NXNY),V(NXNY),W(NXNY)	
277	DIMENSION U1W(NY),U2W(NY),U3W(NY)	
278	DIMENSION ITBS(NX),ITBN(NX),JTBE(NY),JTBW(NY)	
279	C	
280	C	
281	NI=NIM+1	
282	NJ=NJM+1	
283	DO 10 J=2,NJM	
284	IJ=J	

Oct 12 1996 16:36	asmod_2d	Page 5
285	IJM=IJ-1	
286	IPJ=IJ+NJ	
287	U1W(IJ)=U(IJ)	
288	U2W(IJ)=V(IJ)	
289	U3W(IJ)=W(IJ)	
290	CONTINUE	
291	C	
292	DO 101 I=2,NIM	
293	J=1	
294	IJ=IMNJ(I)+J	
295	UN=U(IJ)	
296	VN=V(IJ)	
297	WN=W(IJ)	
298	DO 102 J=2,NJM	
299	IJ=IMNJ(I)+J	
300	IPJ=IJ+NJ	
301	IMJ=IJ-NJ	
302	IJP=IJ+1	
303	IJM=IJ-1	
304	FXE=FX(IJ)	
305	FXW=1.-FXE	
306	FYN=FY(IJ)	
307	FYS=1.-FYN	
308	RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))	
309	DXEW=HAF*(X(IJ)-X(IMJ)+X(IJM)-X(IMJ-1))	
310	DXNS=HAF*(X(IJ)-X(IJM)+X(IMJ)-X(IMJ-1))	
311	DXEW=HAF*(Y(IJ)-Y(IMJ)+Y(IJM)-Y(IMJ-1))	
312	DYNS=HAF*(Y(IJ)-Y(IJM)+Y(IMJ)-Y(IMJ-1))	
313	C	
314	US=UN	
315	VS=VN	
316	WS=WN	
317	C	
318	UN=U(IJ)*FYS+U(IJP)*FYN	
319	VN=V(IJ)*FYS+V(IJP)*FYN	
320	WN=W(IJ)*FYS+W(IJP)*FYN	
321	C	
322	UE=U(IJ)*FXW+U(IPJ)*FXE	
323	VE=V(IJ)*FXW+V(IPJ)*FXE	
324	WE=W(IJ)*FXW+W(IPJ)*FXE	
325	C	
326	DUEW=UE-U1W(J)	
327	DUNS=UN-US	
328	DVEW=VE-U2W(J)	
329	DVNS=VN-VS	
330	DWEW=WE-U3W(J)	
331	DNWS=WN-WS	
332	C	
333	DUDX(IJ)=(DUEW*DYNS-DUNS*DXEW)/ARE(IJ)	
334	DUDY(IJ)=(DUNS*DXEW-DUEW*DXNS)/ARE(IJ)	
335	DVDX(IJ)=(DVEW*DYNS-DVNS*DXEW)/ARE(IJ)	
336	DVDY(IJ)=(DVNS*DXEW-DVEW*DXNS)/ARE(IJ)	
337	DWDY(IJ)=(DWEW*DYNS-DWNS*DXEW)/ARE(IJ)	
338	DWDY(IJ)=(DWNS*DXEW-DWEW*DXNS)/ARE(IJ)	
339	C	
340	C	
341	if(iter.eq.1) then	
342	P11(IJ)=-2.*(U2(IJ)*DUDX(IJ)+UV(IJ)*DUDY(IJ))	
343	P22(IJ)=-2.*(UV(IJ)*DVDX(IJ)+V2(IJ)*DVDY(IJ))	
344	IF(ICAL.EQ.1) P22(IJ)=P22(IJ)+2.*VW(IJ)*W(IJ)/RP	
345	P33(IJ)=0	
346	IF(AKSI) P33(IJ)=-2.*W2(IJ)*V(IJ)/RP	
347	IF(ICAL.EQ.1) P33(IJ)=P33(IJ)-2.*(UW(IJ)*DWDY(IJ)+	
348	& VW(IJ)*DWDY(IJ))	
349	#	
350	UV(IJ)*(DUDX(IJ)+DVDY(IJ))	
351	IF(ICAL.EQ.1) P12(IJ)=P12(IJ)+UW(IJ)*W(IJ)/RP	
352	P13(IJ)=0	
353	P23(IJ)=0	
354	IF(ICAL.EQ.1) THEN	
355	P13(IJ)=-2.*(U2(IJ)*DWDX(IJ)+UV(IJ)*DWDY(IJ)+	

Oct 12 1996 16:36	asmod_2d	Page 6
356	&	VW(IJ)*DUDY(IJ)+UW(IJ)*DUDX(IJ))
357	P23(IJ)=-	(UV(IJ)*DWDX(IJ)+UW(IJ)*DVDY(IJ)+
358	&	V2(IJ)*DWDY(IJ)+VW(IJ)*DVDY(IJ)-
359	&	W2(IJ)*W(IJ)/RP)
360	ENDIF	
361	IF(AKSI) THEN	
362	P13(IJ)=P13(IJ)-UW(IJ)*V(IJ)/RP	
363	P23(IJ)=P23(IJ)-VW(IJ)*V(IJ)/RP	
364	END IF	
365	GEN(IJ)=0.5*ABS(P11(IJ)+P22(IJ)+P33(IJ))	
366	end if	
367	C	
368	U1W(J)=UE	
369	U2W(J)=VE	
370	U3W(J)=WE	
371	C	
372	CONTINUE	
373	101	RETURN
374		
375	END	
376	C	
377	C	
378		SUBROUTINE CALUIUJ (ICAL, AKSI, R, U, V, W, DEN, TE, ED, VIS, VISCOS,
379	&	ITBS, ITBN, JTBE, JTBW, WREFON, iter)
380	C	
381	INCLUDE 'gridparam.h'	
382	INCLUDE 'asm.h'	
383	DIMENSION PHI(NXNY), FXWW(NY), DW(NY)	
384	C	
385	DIMENSION U(NXNY), V(NXNY), W(NXNY), TE(NXNY), ED(NXNY),	
386	#	DEN(NXNY), R(NXNY), A(6,6), B(6), VIS(NXNY)
387	DIMENSION ITBS(NX), ITBN(NX), JTBE(NY), JTBW(NY)	
388	C	
389	NI=NI+1	
390	NJ=NJ+1	
391	C	
392	C	-----CALCULATES ALGEBRAIC STRESS EQUATIONS IN THE FORM
393	C	A11*U2(IJ)+A12*V2(IJ)+A13*W2(IJ)+A14*UV(IJ)+A15*VW(IJ)+A16*UW(IJ)=
394	C	
395	C	RELTA=0.5
396	FERTA1=0.	
397	FERTA2=0.	
398	DO 10 I=2,NIM	
399	DO 10 J=2,NJM	
400	IJ=IMNJ(I)+J	
401	RP=QTR*(R(IJ)+R(IJ-1)+R(IJ-NJ)+R(IJ-NJ-1))	
402	EDK=ED(IJ)/(TE(IJ)+SMALL)	
403	AUX=(1.-C2)/(C1*ED(IJ)+GEN(IJ)/DEN(IJ)-ED(IJ))	
404	AUX=1./(AUX+TE(IJ)+SMALL)	
405	C	ORDER OF VARIABLE U2, V2, W2, UV, VW, UW
406	A(1,1)=1.5*AUX+2.*DUDX(IJ)	
407	A(1,2)=-DVDY(IJ)	
408	A(1,3)=0.	
409	IF(AKSI) A(1,3)=A(1,3)-V(IJ)/RP	
410	A(1,4)=-DVDX(IJ)+2.*DUDY(IJ)	
411	A(1,5)=-DWDY(IJ)	
412	IF(ICAL.EQ.1) A(1,5)=A(1,5)+W(IJ)/RP	
413	A(1,6)=-DWDX(IJ)	
414	C	
415	A(2,1)=-DUDX(IJ)	
416	A(2,2)=1.5*AUX+2.*DVDY(IJ)	
417	A(2,3)=0.	
418	IF(AKSI) A(2,3)=A(2,3)-V(IJ)/RP	
419	A(2,4)=-DUDY(IJ)+2.*DVDX(IJ)	
420	A(2,5)=-DWDY(IJ)	
421	IF(ICAL.EQ.1) A(2,5)=A(2,5)-2.*W(IJ)/RP	
422	A(2,6)=-DWDX(IJ)	
423	C	
424	A(3,1)=-DUDX(IJ)	
425		

Oct 12 1996 16:36	asmod_2d	Page 8
497	UV(IJ)=B(4)*RELT*(1.-RELT)*UV(IJ)	
498	VW(IJ)=B(5)*RELT*(1.-RELT)*VW(IJ)	
499	UW(IJ)=B(6)*RELT*(1.-RELT)*UW(IJ)	
500 C		
501	TAUMAX=2.*DEN(IJ)*TE(IJ)	
502	TAUMIN=0.	
503	U2(IJ)=AMIN1(U2(IJ),TAUMAX)	
504	U2(IJ)=AMAX1(U2(IJ),TAUMIN)	
505	V2(IJ)=AMIN1(V2(IJ),TAUMAX)	
506	V2(IJ)=AMAX1(V2(IJ),TAUMIN)	
507	W2(IJ)=AMIN1(W2(IJ),TAUMAX)	
508	W2(IJ)=AMAX1(W2(IJ),TAUMIN)	
509 C		
510 C		
511	P11(IJ)=-2.*U2(IJ)*DUDX(IJ)+UV(IJ)*DUDY(IJ)	
512	P22(IJ)=-2.*UV(IJ)*DUDX(IJ)+V2(IJ)*DUDY(IJ)	
513	IF(ICAL.EQ.1) P22(IJ)=P22(IJ)+2.*VW(IJ)*W(IJ)/RP	
514	P33(IJ)=0.	
515	IF(AKSI) P33(IJ)=-2.*W2(IJ)*V(IJ)/RP	
516	IF(ICAL.EQ.1) P33(IJ)=P33(IJ)-2.*UW(IJ)*DWDY(IJ)+	
517	& VW(IJ)*DWDY(IJ)+	
518	* P12(IJ)=-U2(IJ)*DUDX(IJ)+V2(IJ)*DUDY(IJ)+	
519	& UV(IJ)*DUDX(IJ)+DUDY(IJ))	
520	IF(ICAL.EQ.1) P12(IJ)=P12(IJ)+UW(IJ)*W(IJ)/RP	
521	P23(IJ)=0.	
522	IF(ICAL.EQ.1) THEN	
523	P13(IJ)=-U2(IJ)*DWDY(IJ)+UV(IJ)*DWDY(IJ)+	
524	& VW(IJ)*DUDY(IJ)+UW(IJ)*DUDX(IJ)	
525	& P23(IJ)=-UW(IJ)*DWDY(IJ)+UW(IJ)*DWDY(IJ)+	
526	& V2(IJ)*DWDY(IJ)+VW(IJ)*DWDY(IJ)-	
527	& W2(IJ)*W(IJ)/RP	
528	ENDIF	
529	IF(AKSI) THEN	
530	P13(IJ)=P13(IJ)-UW(IJ)*V(IJ)/RP	
531	P23(IJ)=P23(IJ)-VW(IJ)*V(IJ)/RP	
532	END IF	
533	GEN(IJ)=0.5*ABS(P11(IJ)+P22(IJ)+P33(IJ))	
534		
535 C		
536 10	CONTINUE	
537 C		
538 C---	MODIFY GEN-TERMS CLOSE TO A WALL	
539 C		
540	CALL MODPIJ (ITBS,ITBN,JTBW,JTBE)	
541 C		
542 C		
543	DO I=1,NI	
544	IJ=IMNJ(I)+1	
545	U2(IJ)=U2(IJ)+1	
546	V2(IJ)=V2(IJ)+1	
547	W2(IJ)=W2(IJ)+1	
548	UV(IJ)=UV(IJ)+1	
549	VW(IJ)=VW(IJ)+1	
550	UW(IJ)=UW(IJ)+1	
551	IJ=IMNJ(I)+NJ	
552	U2(IJ)=U2(IJ)-1	
553	V2(IJ)=V2(IJ)-1	
554	W2(IJ)=W2(IJ)-1	
555	UV(IJ)=UV(IJ)-1	
556	VW(IJ)=VW(IJ)-1	
557	UW(IJ)=UW(IJ)-1	
558	END DO	
559	DO J=2,NJW	
560	IJ=IMNJ(1)+J	
561	U2(IJ)=U2(IJ)+NJ	
562	V2(IJ)=V2(IJ)+NJ	
563	W2(IJ)=W2(IJ)+NJ	
564	UV(IJ)=UV(IJ)+NJ	
565	VW(IJ)=VW(IJ)+NJ	
566	UW(IJ)=UW(IJ)+NJ	
567	IJ=IMNJ(NI)+J	

Oct 12 1996 16:36	asmod_2d	Page 7
426	A(3,2)=-DUDY(IJ)	
427	A(3,3)=1.5*AUX	
428	IF(AKSI) A(3,3)=A(3,3)+2.*V(IJ)/RP	
429	A(3,4)=-DUDY(IJ)-DUDX(IJ)	
430	A(3,5)=2.*DWDY(IJ)	
431	IF(ICAL.EQ.1) A(3,5)=A(3,5)+W(IJ)/RP	
432	A(3,6)=2.*DWDX(IJ)	
433 C		
434	A(4,1)=DUDX(IJ)	
435	A(4,2)=DUDY(IJ)	
436	A(4,3)=0.	
437	A(4,4)=AUX+DUDX(IJ)+DUDY(IJ)	
438	A(4,5)=0.	
439	A(4,6)=0.	
440	IF(ICAL.EQ.1) A(4,6)=A(4,6)-W(IJ)/RP	
441 C		
442	A(5,1)=0.	
443	A(5,2)=DWDY(IJ)	
444	A(5,3)=0.	
445	IF(ICAL.EQ.1) A(5,3)=A(5,3)-W(IJ)/RP	
446	A(5,4)=DWDX(IJ)	
447	A(5,5)=AUX+DUDY(IJ)	
448	IF(AKSI) A(5,5)=A(5,5)+V(IJ)/RP	
449	A(5,6)=DUDX(IJ)	
450 C		
451	A(6,1)=DWDX(IJ)	
452	A(6,2)=0.	
453	A(6,3)=0.	
454	A(6,4)=DWDY(IJ)	
455	A(6,5)=DUDY(IJ)	
456	A(6,6)=AUX+DUDX(IJ)	
457	IF(AKSI) A(6,6)=A(6,6)+V(IJ)/RP	
458 C		
459	B(1)=AUX+DEN(IJ)*TE(IJ)	
460	B(2)=B(1)	
461	B(3)=B(1)	
462	B(4)=0.	
463	B(5)=0.	
464	B(6)=0.	
465	IF(WREFON.EQ.1.) THEN	
466	B1=1./(1.-C2)	
467	B2=C1W*ED(IJ)/(TE(IJ)+SMALL)	
468	B3=C2*C2W	
469	Fw1=-2.*U2(IJ)*FUNK(IJ)-UV(IJ)*FUNK(IJ)+V2(IJ)*FUNK(IJ)	
470	Fw2=-2.*P11(IJ)-2./3.*GEN(IJ)*FUNK(IJ)+P12(IJ)*FUNK(IJ)	
471	# B(1)=B(1)+B1*1.5*(B2*Fw1+B3*Fw2)	
472	Fw1=U2(IJ)*FUNK(IJ)-UV(IJ)*FUNK(IJ)+V2(IJ)*FUNK(IJ)	
473	Fw2=-2.*P11(IJ)-2./3.*GEN(IJ)*FUNK(IJ)+P12(IJ)*FUNK(IJ)	
474	# B(2)=B(2)+B1*1.5*(B2*Fw1+B3*Fw2)	
475	Fw1=U2(IJ)*FUNK(IJ)+2.*UV(IJ)*FUNK(IJ)+V2(IJ)*FUNK(IJ)	
476	Fw2=-2.*P11(IJ)-2./3.*GEN(IJ)*FUNK(IJ)+P12(IJ)*FUNK(IJ)	
477	# B(3)=B(3)+B1*1.5*(B2*Fw1+B3*Fw2)	
478	Fw1=U2(IJ)+V2(IJ)*FUNK(IJ)-UV(IJ)*FUNK(IJ)+FUNK(IJ)*FUNK(IJ)	
479	Fw2=-2.*P11(IJ)+P22(IJ)-4./3.*GEN(IJ)*FUNK(IJ)+P12(IJ)*FUNK(IJ)+	
480	# P12(IJ)*FUNK(IJ)+FUNK(IJ)	
481	B(4)=B(4)+B1*(B2*Fw1+B3*Fw2)	
482	Fw1=-UW(IJ)*FUNK(IJ)+P23(IJ)*FUNK(IJ)	
483	B(5)=B(5)+B1*(B2*Fw1+B3*Fw2)	
484	Fw1=-UW(IJ)*FUNK(IJ)-VW(IJ)*FUNK(IJ)	
485	B(6)=B(6)+B1*(B2*Fw1+B3*Fw2)	
486	Fw1=-UW(IJ)*FUNK(IJ)-VW(IJ)*FUNK(IJ)	
487	Fw2=P13(IJ)*FUNK(IJ)+P23(IJ)*FUNK(IJ)	
488	Fw1=-UW(IJ)*FUNK(IJ)-VW(IJ)*FUNK(IJ)	
489	Fw2=P13(IJ)*FUNK(IJ)+P23(IJ)*FUNK(IJ)	
490	B(6)=B(6)+B1*(B2*Fw1+B3*Fw2)	
491	END IF	
492 C		
493	CALL SOLV(A,B,6)	
494	U2(IJ)=B(1)*RELT*(1.-RELT)*U2(IJ)	
495	V2(IJ)=B(2)*RELT*(1.-RELT)*V2(IJ)	
496	W2(IJ)=B(3)*RELT*(1.-RELT)*W2(IJ)	

Oct 12 1996 16:36	asmod_2d	Page 10
639	SNSW(J)=0.0	
640	FYMW(J)=1.0	
641	IF(JTBW(J).EQ.3.OR.JTBW(J).EQ.4) GO TO 10	
642	DXKS=QTR*(X(IJ+NUJ)+X(IJ+NUJ-1)-X(IJ)-X(IJ-1))	
643	DXKS=QTR*(Y(IJ+NUJ)+Y(IJ+NUJ-1)-Y(IJ)-Y(IJ-1))	
644	SNSW(J)=-(GAMEU2*DYKS+DYE+GAMEV2*DXKS+DXE)*(PHINE-PHISE)+	
645	& 2.*GAMEUV*DXE*DYE*(PHI(IPJ)-PHI(IJ))-	
646	& GAMEUV*(DXKS+DYE+DYKS+DXE)*(PHINE-PHISE))	
647	IF(LAY2.AND.FMU(IJ).LT.0.95) THEN	
648	SNSW(J)=GAME/AREE*(DXKS+DXE+DYKS+DYE)*(PHINE-PHISE)	
649	END IF	
650	10 CONTINUE	
651	C	
652	DO 100 I=2,NIM	
653	J=1	
654	IJ=IMNJ(I)+J	
655	IMJ=IJ-NUJ	
656	IJP=IJ+1	
657	FXE=FX(IJ)	
658	FYN=1.-FX(IJ)	
659	FYN=FY(IJ)	
660	FYS=1.0-FYN	
661	AREN=HAF*(ARE(IJ)+ARE(IJP))	
662	DXN=X(IJ)-X(IMJ)	
663	DYN=Y(IJ)-Y(IMJ)	
664	DXET=QTR*(X(IJP)+X(IJP-NUJ)-X(IJ)-X(IMJ))	
665	DYET=QTR*(Y(IJP)+Y(IJP-NUJ)-Y(IJ)-Y(IMJ))	
666	GAMNU2=0.0	
667	GAMNV2=0.0	
668	GAMNUV=0.0	
669	CKK=CK	
670	IF(IPHI.EQ.2) CKK=CE	
671	IF(ED(IJ).NE.0.0) THEN	
672	TERM=HAF*TE(IJ)/ED(IJ)*CKK*(R(IJ)+R(IMJ))	
673	GAMNU2=TERM*U2(IJ)/AREN	
674	GAMNV2=TERM*V2(IJ)/AREN	
675	GAMNUV=TERM*UV(IJ)/AREN	
676	ENDIF	
677	DN=GAMNU2*DXN**2+GAMNV2*DXN**2	
678	IF(LAY2.AND.FMU(IJ).LT.0.95) THEN	
679	VIST=VIS(IJ)-VISCOS	
680	GAMN=HAF*(VISCOS+VIST*PTINVP)*(R(IJ)+R(IJ-NUJ))	
681	DN=GAMN/AREN*(DXN**2+DYN**2)	
682	END IF	
683	FYSS=1.0	
684	PHINE=PHI(IJ+NUJ)*FXE+PHI(IJ)*FYW	
685	SEWN=0.0	
686	IF(ITBS(I).EQ.3.OR.ITBS(I).EQ.4) GO TO 110	
687	SEWN=-((GAMNU2*DXN*DYN*(PHI(IJP)-PHI(IJ))-	
688	& 2.*GAMNUV*DXN*DYN*(PHI(IJP)-PHI(IJ))-	
689	& GAMNUV*(DXET*DYN+DYET*(PHINE-PHINW(J))	
690	IF(LAY2.AND.FMU(IJ).LT.0.95) THEN	
691	SEWN=-GAMN/AREN*(DXN*DXET+DYN*DYET)*(PHINE-PHINW(J))	
692	END IF	
693	110 CONTINUE	
694	PHINW(J)=PHINE	
695	C	
696	C-----THE MAIN LOOP - ASSEMBLY OF COEFFICIENTS AND SOURCES	
697	C	
698	DO 101 J=2,NJM	
699	IJ=IMNJ(I)+J	
700	IPJ=IJ+NUJ	
701	IJP=IPJ+1	
702	IMJ=IJ-NUJ	
703	IJP=IJ+1	
704	IJM=IJ-1	
705	FXE=FX(IJ)	
706	FYN=1.-FXE	
707	FYN=FY(IJ)	
708	FYS=1.-FYN	
709	C	

Oct 12 1996 16:36	asmod_2d	Page 9
568	U2(IJ)=U2(IJ-NUJ)	
569	V2(IJ)=V2(IJ-NUJ)	
570	W2(IJ)=W2(IJ-NUJ)	
571	UV(IJ)=UV(IJ-NUJ)	
572	VW(IJ)=VW(IJ-NUJ)	
573	UW(IJ)=UW(IJ-NUJ)	
574	END DO	
575	RETURN	
576	END	
577	C	
578	C	
579	C-----	
580	SUBROUTINE ACALCKE(PHI,X,Y,FX,FY,ARE,VOL,R,DNS,DNN,DNE,DNW,	
581	& U,V,W,DEN,F1,F2,VIS,VISCOS,TE,ED,	
582	& ITBS,ITBN,JTBE,JTBW,AKSI,IPHI,RESOR)	
583	C-----	
584	INCLUDE 'gridparam.h'	
585	INCLUDE 'asm.h'	
586	C	
587	DIMENSION PHI(NXNY),FXMW(NY),DW(NY)	
588	C	
589	DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY)	
590	DIMENSION ARE(NXNY),VOL(NXNY),R(NXNY),VIS(NXNY)	
591	DIMENSION F1(NXNY),F2(NXNY),TE(NXNY),ED(NXNY)	
592	DIMENSION DNS(NX),DNN(NX),DNW(NY),DNE(NY)	
593	DIMENSION U(NXNY),V(NXNY),W(NXNY),DEN(NXNY)	
594	DIMENSION ITBS(NX),ITBN(NX),JTBW(NY),JTBE(NY)	
595	DIMENSION PHINW(NY),SNSW(NY),RESOR(2)	
596	C	
597	NI=NIM+1	
598	NU=NUM+1	
599	C-----INITIALISATION OF TEMPORALLY STORED VARIABLES	
600	C	
601	URFPHI=1./URFKE(IPHI)	
602	PTINVP=1./PTWKE(IPHI)	
603	C	
604	C	
605	IJ=1	
606	PHINE=PHI(IJ)	
607	PHINW(IJ)=PHINE	
608	DO 10 J=2,NJM	
609	IJ=J	
610	IJM=IJ-1	
611	IPJ=IJ+NUJ	
612	IJP=IPJ+1	
613	FXE=FX(IJ)	
614	FYN=1.-FXE	
615	FYN=FY(IJ)	
616	FYS=1.0-FYN	
617	C	
618	AREE=HAF*(ARE(IJ)+ARE(IPJ))	
619	DXE=X(IJ)-X(IJM)	
620	DYE=Y(IJ)-Y(IJM)	
621	CC	
622	CKK=CK	
623	IF(IPHI.EQ.2) CKK=CE	
624	IF(ED(IJ).NE.0.0) THEN	
625	TERM=HAF*TE(IJ)/ED(IJ)*CKK*(R(IJ)+R(IJM))	
626	GAMU2=TERM*U2(IJ)/AREE	
627	GAMV2=TERM*V2(IJ)/AREE	
628	GAMUV=TERM*UV(IJ)/AREE	
629	ENDIF	
630	DW(J)=GAMEU2*DXE**2+GAMEV2*DXE**2	
631	IF(LAY2.AND.FMU(IJ).LT.0.95) THEN	
632	VIST=VIS(IJ)-VISCOS	
633	GAM=HAF*(VISCOS+VIST*PTINVP)*(R(IJ)+R(IJ-1))	
634	DW(J)=GAME/AREE*(DXE**2+DYE**2)	
635	END IF	
636	PHISE=PHINE	
637	PHINE=PHI(IJ+1)*FY(IJ)+PHI(IJ)*(1.-FY(IJ))	
638	PHINW(J)=PHINE	

Oct 12 1996 16:36

asmod_2d

Page 11

```

710 DYE=X(IJ)-X(IJM)
711 DYE=Y(IJ)-Y(IJM)
712 DXN=X(IJ)-X(IMJ)
713 DYN=Y(IJ)-Y(IMJ)
714 DXKS=QTR*(X(IPJ)-X(IMJ)+X(IPJ-1)-X(IMJ-1))
715 DYKS=QTR*(Y(IPJ)-Y(IMJ)+Y(IPJ-1)-Y(IMJ-1))
716 DXT=QTR*(X(IJP)-X(IJM)+X(IJP-NJ)-X(IJM-NJ))
717 DYET=QTR*(Y(IJP)-Y(IJM)+Y(IJP-NJ)-Y(IJM-NJ))
718 AREE=HAF*(AREE(IJ)+AREE(IPJ))
719 AREN=HAF*(AREE(IJ)+AREE(IPJ))
720 CKK=CK
721 IF (I=1) EQ.2 CKK=CE
722 GAMEU2=HAF*CKK/AREE*(TE(IJ)/(ED(IJ)+SMALL)*U2(IJ)*FXW+
723 & TE(IPJ)/(ED(IPJ)+SMALL)*U2(IPJ)*FXE)*(R(IJ)+R(IJM))
724 GAMEV2=HAF*CKK/AREE*(TE(IJ)/(ED(IJ)+SMALL)*V2(IJ)*FXW+
725 & TE(IPJ)/(ED(IPJ)+SMALL)*V2(IPJ)*FXE)*(R(IJ)+R(IJM))
726 GAMEUV=HAF*CKK/AREE*(TE(IJ)/(ED(IJ)+SMALL)*UV(IJ)*FXW+
727 & TE(IPJ)/(ED(IPJ)+SMALL)*UV(IPJ)*FXE)*(R(IJ)+R(IJM))
728 C
729 GAMNU2=HAF*CKK/AREN*(TE(IJ)/(ED(IJ)+SMALL)*U2(IJ)*FYS+
730 & TE(IPJ)/(ED(IPJ)+SMALL)*U2(IPJ)*FYN)*(R(IJ)+R(IMJ))
731 GAMNV2=HAF*CKK/AREN*(TE(IJ)/(ED(IJ)+SMALL)*V2(IJ)*FYS+
732 & TE(IPJ)/(ED(IPJ)+SMALL)*V2(IPJ)*FYN)*(R(IJ)+R(IMJ))
733 GAMNUV=HAF*CKK/AREN*(TE(IJ)/(ED(IJ)+SMALL)*UV(IJ)*FYS+
734 & TE(IPJ)/(ED(IPJ)+SMALL)*UV(IPJ)*FYN)*(R(IJ)+R(IMJ))
735 C
736 IF (LAY2.AND.FMU(IJ).LT.0.95) THEN
737 VISE=VIS(IJ)*FXW+VIS(IPJ)*FXE
738 VISE=VISE-VISCOS
739 GAME=HAF*(VISCOS+VISE*PRTNVP)*(R(IJ)+R(IJM))
740 VISN=VIS(IJ)*FYS+VIS(IPJ)*FYN
741 VISNT=VISN-VISCOS
742 GAMN=HAF*(VISCOS+VISNT*PRTNVP)*(R(IJ)+R(IMJ))
743 END IF
744 DS=DN
745 DE=GAMNU2*DYE**2+GAMEV2*DXE**2
746 DN=GAMNU2*DYN**2+GAMNV2*DXN**2
747 IF (LAY2.AND.FMU(IJ).LT.0.95) THEN
748 DE=GAME/AREE*(DXE**2+DYE**2)
749 DN=GAMN/AREN*(DXN**2+DYN**2)
750 END IF
751 C
752 C LINEAR UPWIND DIFFERENCING
753 C
754 AEE=AMINI(F1(IJ),0.0)*FX(IPJ)*GKE
755 AEW=AMAX1(F1(IMJ),0.0)*(1.0-FXW(J))*GKE
756 AEI=-AMINI(F1(IMJ),0.0)*FXE*GKE
757 AWI=AMAX1(F1(IJ),0.0)*(1.0-FX(IMJ))*GKE
758 ANN=AMINI(F2(IJ),0.0)*FY(IPJ)*GKE
759 ASS=-AMAX1(F2(IJM),0.0)*(1.0-FYSS)*GKE
760 ANI=-AMINI(F2(IJM),0.0)*FYN*GKE
761 ASI=AMAX1(F2(IJ),0.0)*(1.0-FY(IMJ))*GKE
762 C
763 AW(IJ)=DW(J)+AMAX1(F1(IMJ),0.0)-AEW
764 AE(IJ)=DE-AMINI(F1(IJ),0.0)-AEI
765 AS(IJ)=DS-AMAX1(F2(IJM),0.0)-ASS
766 AN(IJ)=DN-AMINI(F2(IJ),0.0)-ANN
767 C
768 DXKS=QTR*(X(IPJ)-X(IMJ)+X(IPJ-1)-X(IMJ-1))
769 DYKS=QTR*(Y(IPJ)-Y(IMJ)+Y(IPJ-1)-Y(IMJ-1))
770 DXT=QTR*(X(IJP)-X(IJM)+X(IJP-NJ)-X(IJM-NJ))
771 DYET=QTR*(Y(IJP)-Y(IJM)+Y(IJP-NJ)-Y(IJM-NJ))
772 C
773 PHISE=PHINE
774 PHINE=(PHI(IJ)+FYS+PHI(IPJ)*FYN)*FXW+
775 & (PHI(IPJ)+FYS+PHI(IPJ+1)*FYN)*FXE
776 SEWS=SEWN
777 SEWN=-((GAMNU2*DYN*DYET+GAMNV2*DXN*DXT)*
778 & (PHINE-PHINW(J))-GAMNUV*(DXE*DYN+DXN*DYET)*
779 & (PHINE-PHINW(J))+2.*GAMNUV*DXN*DYN*
780 & (PHI(IPJ)-PHI(IJ)))

```

Oct 12 1996 16:36

asmod_2d

Page 12

```

781 C
782 SNSE=-((GAMEU2*DYKS*DYE+GAMEV2*DXKS*DXE)*
783 & (PHINE-PHISE)+2.*GAMNUV*DYE*DYE*
784 & (PHI(IPJ)-PHI(IJ))-GAMEUV*(DXKS*DYE+DYKS*DXE)*
785 & (PHINE-PHISE))
786 IF (LAY2.AND.FMU(IJ).LT.0.95) THEN
787 SEWN=-GAMN/AREN*(DXN*DXT+DYN*DYET)*(PHINE-PHINW(J))
788 SNSE=-GAME/AREE*(DXKS*DXE+DYKS*DYE)*(PHINE-PHISE)
789 END IF
790 C
791 IF (I=1) EQ.NIM.AND.(JTBE(J).EQ.3.OR.JTBE(J).EQ.4)) SNSE=0.
792 IF (J.EQ.NJM.AND.(ITBN(I).EQ.3.OR.ITBN(I).EQ.4)) SEWN=0.
793 C
794 C LINEAR UPWIND DIFFERENCING
795 C
796 IMJ1=IMJ-NJ
797 IMJ=MAX(1,IMJ1)
798 APV(IJ)=SNSE-SNSW(J)+SEWN-SEWS
799 APV(IJ)=APV(IJ)+AEE*PHI(IPJ+NJ)+AMW*PHI(IMJ)+ANN
800 & PHI(IPJ+1)+ASS*PHI(IJM-1)
801 & +AEI*PHI(IPJ)+AWI*PHI(IMJ)+ANI*PHI(IJP)+AS1*PHI(IJM)
802 APU(IJ)=AEE*AMW+ANN+ASS*AEI+AWI+ANI+AS1
803 C
804 GO TO (120,130) IPHI
805 C
806 C
807 C-----TURBULENT KINETIC ENERGY SOURCE TERMS
808 C
809 120 CONTINUE
810 SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
811 BP(IJ)=APU(IJ)+VOL(IJ)*DEN(IJ)*ED(IJ)/(TE(IJ)+SMALL)
812 C
813 GO TO 150
814 C
815 C-----DISSIP.OF TURB. KIN. ENERGY SOURCE TERMS
816 C
817 130 CONTINUE
818 C
819 SU(IJ)=APV(IJ)+VOL(IJ)*CD1*ED(IJ)*ABS(GEN(IJ))/
820 & (TE(IJ)+SMALL)
821 BP(IJ)=APU(IJ)+VOL(IJ)*CD2*DEN(IJ)*ED(IJ)/
822 & (TE(IJ)+SMALL)
823 C
824 150 CONTINUE
825 C
826 SNSW(J)=SNSE
827 PHINW(J)=PHINE
828 FYSS=FY(IJM)
829 FXW(J)=FX(IMJ)
830 DW(J)=DE
831 C
832 101 CONTINUE
833 100 CONTINUE
834 C
835 C-----PROBLEM MODIFICATIONS - BOUNDARY CONDITIONS
836 C
837 CALL MODPHI (PHI, IPHI, X, Y, FX, FY, ARE, VOL, R,
838 & DEN, TE, ED, ITES, ITEN, JTBE, JTBN,
839 & DNS, DNN, DNE, DNM)
840 C
841 IF (IPHI.EQ.2.AND.LAY2) THEN
842 DO 30 I=2,NIM
843 IF (ITES(I).EQ.4.OR.ITBN(I).EQ.4) THEN
844 DO 310 J=2,NJM
845 IJ=IMNJ(I)+J
846 IF (FMU(IJ).LT.0.95) THEN
847 SU(IJ)=GREAT*ED(IJ)
848 BP(IJ)=GREAT
849 END IF
850 CONTINUE
851 ENDDIF

```

Oct 12 1996 16:36	asmod_2d	Page 13
852 C	CONTINUE	
853 C	DO 40 J=2,NJM	
854	IF (JTBW(J).EQ.4.OR.JTBE(J).EQ.4) THEN	
855	DO 410 I=2,NIM	
856	IJ=IMNJ(I)+J	
857	IF(FMU(IJ).LT.0.95) THEN	
858	SU(IJ)=GREAT*ED(IJ)	
859	BP(IJ)=GREAT	
860	END IF	
861	CONTINUE	
862 410	ENDIF	
863	CONTINUE	
864 40	ENDIF	
865 C		
866 C	DO 200 I=2,NJM	
867 C	DO 201 J=2,NJM	
868	IJ=IMNJ(I)+J	
869	AP(IJ)=AW(IJ)+AE(IJ)+AN(IJ)+AS(IJ)+BP(IJ)	
870	AP(IJ)=AP(IJ)+URFPHI	
871	SU(IJ)=SU(IJ)+(1.-URFKE(IPHI))*AP(IJ)*PHI(IJ)	
872	CONTINUE	
873 201	CONTINUE	
874 200	CONTINUE	
875 C		
876 C	SOLVING F.D. EQUATIONS	
877 C		
878 C	CALL ASOLSIP (PHI,IPHI,RESOR)	
879		
880 C	NINJ=NIM	
881	DO IJ=1,NINJ	
882	IF(PHI(IJ).LT.0.) PHI(IJ)=ABS(PHI(IJ))	
883	END DO	
884	RETURN	
885	END	
886 C		
887 C		
888 C		
889 C	SUBROUTINE ATWOLAY(X,Y,DEN,TE,ED,VISCOS,ITBS,ITBN,JTBW,JTBE)	
890		
891 C		
892 C	INCLUDE 'gridparam.h'	
893	INCLUDE 'asm.h'	
894		
895 C		
896 C	DIMENSION X(NXNY), Y(NXNY), DEN(NXNY)	
897	DIMENSION TE(NXNY), ED(NXNY)	
898	DIMENSION ITBS(NX), ITBN(NX), JTBE(NY), JTBW(NY)	
899 C		
900 C	NI = NIM + 1	
901	NJ = NJM + 1	
902		
903 C	CMU25=SQRT(SQRT(CMU))	
904	CMU75=CMU25**3	
905	C11=CAPPA/CMU75	
906	AED=2.0*C11	
907	AMU=70.0	
908		
909 C		
910	DO 10 I=2,NIM	
911 C		
912	DO 110 J=2,NJM	
913	DISN=GREAT	
914	DISNW=GREAT	
915	DISNN=GREAT	
916	DISNE=GREAT	
917	IJ=IMNJ(I)+J	
918 C		
919 C	CHECK THE SOUTH BOUNDARY	
920 C	IF(ITBS(I).EQ.4) THEN	
921	IJW=IMNJ(I)+1	
922		

Oct 12 1996 16:36	asmod_2d	Page 14
923	IMJW=IJW-NJ	
924	DXB=X(IJW)-X(IMJW)	
925	DYB=Y(IJW)-Y(IMJW)	
926	XPW=HAF*(X(IJW)+X(IMJW))	
927	YPW=HAF*(Y(IJW)+Y(IMJW))	
928	XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
929	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
930	DXBP=XBP-XPW	
931	DYBP=YBP-YPW	
932	DISN=DELTA(DXB,DYB,DXBP,DYBP)	
933	END IF	
934 C		
935 C	CHECK THE NORTH BOUNDARY	
936 C	IF (ITBN(I).EQ.4) THEN	
937	IJW=IMNJ(I)+NJM	
938	IMJW=IJW-NJ	
939	DXB=X(IJW)-X(IMJW)	
940	DYB=Y(IJW)-Y(IMJW)	
941	XPW=HAF*(X(IJW)+X(IMJW))	
942	YPW=HAF*(Y(IJW)+Y(IMJW))	
943	XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
944	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
945	DXBP=XBP-XPW	
946	DYBP=YBP-YPW	
947	DISNN=DELTA(DXB,DYB,DXBP,DYBP)	
948	END IF	
949 C		
950 C	CHECK WEST BOUNDARY	
951 C	IF (JTBW(J).EQ.4) THEN	
952 C	IJW=J	
953	IMJW=IJW-1	
954	DXB=X(IJW)-X(IMJW)	
955	DYB=Y(IJW)-Y(IMJW)	
956	XPW=HAF*(X(IJW)+X(IMJW))	
957	YPW=HAF*(Y(IJW)+Y(IMJW))	
958	XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
959	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
960	DXBP=XBP-XPW	
961	DYBP=YBP-YPW	
962	DISNW=DELTA(DXB,DYB,DXBP,DYBP)	
963	ENDIF	
964		
965 C	CHECK EAST BOUNDARY	
966	IF (JTBE(J).EQ.4) THEN	
967	IJW=IMNJ(NIM)+J	
968	IMJW=IJW-1	
969	DXB=X(IJW)-X(IMJW)	
970	DYB=Y(IJW)-Y(IMJW)	
971	XPW=HAF*(X(IJW)+X(IMJW))	
972	YPW=HAF*(Y(IJW)+Y(IMJW))	
973	XBP=QTR*(X(IJ)+X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
974	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
975	DXBP=XBP-XPW	
976	DYBP=YBP-YPW	
977	DISNE=DELTA(DXB,DYB,DXBP,DYBP)	
978	ENDIF	
979 C		
980 C		
981	DISN=AMIN1(DISN,DISNN,DISNW,DISNE)	
982	RK=DISN*DEN(IJ)*SQRT(TE(IJ))/VISCOS	
983	RT=DEN(IJ)*TE(IJ)*TE(IJ)/(VISCOS*ED(IJ))	
984	ALMU=C11*DISN*(1.0-EXP(-RK/AMU))	
985	ALED=C11*DISN*(1.0-EXP(-RK/AED))	
986	FMU(IJ)=ALMU/ALED*SMALL	
987	FMU(IJ)=AMIN1(FMU(IJ),1.)	
988	IF(FMU(IJ).GE.0.95) FMU(IJ)=1.	
989	IF(FMU(IJ).LT.0.95) THEN	
990	ED(IJ)=SQRT(TE(IJ))*3/ALED	
991	VIS2(IJ)=VISCOS+DEN(IJ)*CMU*SQRT(TE(IJ))*ALMU	
992	END IF	
993	110 CONTINUE	

Oct 12 1996 16:36 asmod_2d Page 15

```

994 10 CONTINUE
995 C
996 RETURN
997 END
998 C
999 C-----
1000 SUBROUTINE MODPIJ (ITBS, ITBN, JTBN, JTBE)
1001 C-----
1002 INCLUDE 'gridparam.h'
1003 INCLUDE 'asm.h'
1004 DIMENSION ITBS(NX), ITBN(NX), JTBN(NY), JTBE(NY)
1005 NI=NI*1
1006 NJ=NJ*1
1007 C
1008 C-- SOUTHE WALL
1009 C
1010 DO 1110 I=2,NIM
1011 IJ=IMNJ(I)+2
1012 IF (ITBS(I).EQ.4) THEN
1013 GEN(IJ)=GENTS(I)
1014 IF (.NOT.LAY2) GEN(IJ)=GENTS(I)
1015 ENDIF
1016 C
1017 C--- NORTH WALL
1018 C
1019 IJ=IMNJ(I)+NJ
1020 IF (ITBN(I).EQ.4) THEN
1021 GEN(IJ)=GENTN(I)
1022 IF (.NOT.LAY2) GEN(IJ)=GENTN(I)
1023 ENDIF
1024 C
1025 1110 CONTINUE
1026 C
1027 C--- WEST WALL
1028 C
1029 DO 1120 J=2,NJM
1030 IJ=IMNJ(2)+J
1031 IF (JTBN(J).EQ.4) THEN
1032 GEN(IJ)=GENTW(J)
1033 IF (.NOT.LAY2) GEN(IJ)=GENTW(J)
1034 ENDIF
1035 C
1036 C--- EAST WALL
1037 C
1038 IJ=IMNJ(NIM)+J
1039 IF (JTBE(J).EQ.4) THEN
1040 GEN(IJ)=GENTEE(J)
1041 IF (.NOT.LAY2) GEN(IJ)=GENTEE(J)
1042 ENDIF
1043 C
1044 1120 CONTINUE
1045 C
1046 RETURN
1047 END
1048 C
1049 C-----
1050 SUBROUTINE MODPHI (PHI, IPHI, X, Y, FX, FY, ARE, VOL, R,
1051 & DEN, TE, ED, ITBS, ITBN, JTBE, JTBN,
1052 & DNS, DNN, DNE, DNW)
1053 C-----
1054 INCLUDE 'gridparam.h'
1055 INCLUDE 'asm.h'
1056 C
1057 C
1058 DIMENSION PHI(NXNY), DEN(NXNY), TE(NXNY), ED(NXNY)
1059 DIMENSION X(NXNY), Y(NXNY), FX(NXNY), FY(NXNY), ARE(NXNY),
1060 # VOL(NXNY), R(NXNY)
1061 DIMENSION ITBS(NX), ITBN(NX), JTBN(NY), JTBE(NY)
1062 DIMENSION DNS(NX), DNN(NX), DNE(NY), DNE(NY)
1063 C
1064 C

```

Oct 12 1996 16:36 asmod_2d Page 16

```

1065 NI=NIM+1
1066 NJ=NJM+1
1067 CMU25=SQRT(SQRT(CMU))
1068 CMU75=CMU25**3
1069 C
1070 GO TO (800,900) IPHI
1071 C
1072 C-----BOUNDARY CONDITIONS FOR KINETIC TURBULENT ENERGY
1073 C
1074 800 CONTINUE
1075 C-----SOUTH BOUNDARY
1076 DO 810 I=2,NIM
1077 IJ=IMNJ(I)+2
1078 GO TO (811,812,813,814) ITBS(I)
1079 811 CONTINUE
1080 SU(IJ)=SU(IJ)+AS(IJ)*TE(IJ-1)
1081 BP(IJ)=BP(IJ)+AS(IJ)
1082 GO TO 815
1083 812 CONTINUE
1084 PHI(IJ-1)=TE(IJ)
1085 GOTO 815
1086 813 CONTINUE
1087 IJ=IJ-1
1088 IPJ=IJ+NJ
1089 IMJ=IJ-NJ
1090 FXE1=FX(IJ)
1091 FXE2=FX(IMJ)
1092 FXW1=1.-FXE1
1093 FXW2=1.-FXE2
1094 DXB=X(IJ)-X(IMJ)
1095 DYB=Y(IJ)-Y(IMJ)
1096 DXBP=QTR*(X(IJ+1)-X(IJ)+X(IMJ+1)-X(IMJ))
1097 DYBP=QTR*(Y(IJ+1)-Y(IJ)+Y(IMJ+1)-Y(IMJ))
1098 FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
1099 DEL=TE(IJ)*(FXW1-FXE2)+TE(IPJ)*FXE1-TE(IMJ)*FXW2
1100 PHI(IJ)=TE(IJ+1)-DEL*FAC
1101 IJ=IJ+1
1102 GOTO 815
1103 814 CONTINUE
1104 IF (.NOT.LAY2) GEN(IJ)=GENTS(I)
1105 SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)
1106 815 CONTINUE
1107 AS(IJ)=0.0
1108 810 CONTINUE
1109 C-----NORTH BOUNDARY
1110 DO 820 I=2,NIM
1111 IJ=IMNJ(I)+NJ
1112 GO TO (821,822,823,824) ITBN(I)
1113 821 CONTINUE
1114 SU(IJ)=SU(IJ)+AN(IJ)*TE(IJ+1)
1115 BP(IJ)=BP(IJ)+AN(IJ)
1116 GO TO 825
1117 822 CONTINUE
1118 PHI(IJ+1)=TE(IJ)
1119 GO TO 825
1120 823 CONTINUE
1121 IJ=IJ+1
1122 IPJ=IJ+NJ
1123 IMJ=IJ-NJ
1124 FXE1=FX(IJ)
1125 FXE2=FX(IMJ)
1126 FXW1=1.-FXE1
1127 FXW2=1.-FXE2
1128 DXB=X(IJ)-X(IMJ)
1129 DYB=Y(IJ)-Y(IMJ)
1130 DXBP=QTR*(X(IJ+1)-X(IJ)+X(IMJ+1)-X(IMJ))
1131 DYBP=QTR*(Y(IJ+1)-Y(IJ)+Y(IMJ+1)-Y(IMJ))
1132 FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)
1133 DEL=TE(IJ)*(FXW1-FXE2)+TE(IPJ)*FXE1-TE(IMJ)*FXW2
1134 PHI(IJ)=TE(IJ-1)-DEL*FAC
1135 IJ=IJ-1

```

Oct 12 1996 16:36	asmod_2d	Page 17
1136	GO TO 825	
1137	824 CONTINUE	
1138	IF (.NOT. LAY2) GEN(IJ)=GENTN(I)	
1139	SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)	
1140	CONTINUE	
1141	AN(IJ)=0.0	
1142	820 CONTINUE	
1143	C-----WEST BOUNDARY	
1144	DO 830 J=2,NJM	
1145	IJ=IMNJ(I)+J	
1146	GO TO (831,832,833,834) JTBW(J)	
1147	831 CONTINUE	
1148	SU(IJ)=SU(IJ)+AW(IJ)*TE(IJ-NJ)	
1149	BP(IJ)=BP(IJ)+AW(IJ)	
1150	GO TO 835	
1151	832 CONTINUE	
1152	PHI(IJ-NJ)=TE(IJ)	
1153	GO TO 835	
1154	833 CONTINUE	
1155	IJ=J	
1156	IJP=IJ+1	
1157	IJM=IJ-1	
1158	FYN1=FY(IJ)	
1159	FYN2=FY(IJM)	
1160	FYS1=1.-FYN1	
1161	FYS2=1.-FYN2	
1162	DYB=Y(IJ)-Y(IJM)	
1163	DXB=X(IJ)-X(IJM)	
1164	DXBP=QTR*(X(IJ)+NJ)-X(IJ)+X(IJM+NJ)-X(IJM)	
1165	DYBP=QTR*(Y(IJ)+NJ)-Y(IJ)+Y(IJM+NJ)-Y(IJM)	
1166	FAC=(DXB+DXBP+DYB+DYBP)/(DXB**2+DYB**2+SMALL)	
1167	DEL=TE(IJ)*(FYS1-FYN2)*TE(IJP)*FYN1-TE(IJM)*FYS2	
1168	PHI(IJ)=TE(IJ+NJ)-DEL*FAC	
1169	IJ=IJ+NJ	
1170	GOTO 835	
1171	834 CONTINUE	
1172	IF (.NOT. LAY2) GEN(IJ)=GENTW(J)	
1173	SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)	
1174	CONTINUE	
1175	AW(IJ)=0.0	
1176	830 CONTINUE	
1177	C-----EAST BOUNDARY	
1178	DO 840 J=2,NJM	
1179	IJ=IMNJ(NIM)+J	
1180	GO TO (841,842,843,844) JTBE(J)	
1181	CONTINUE	
1182	SU(IJ)=SU(IJ)+AE(IJ)*TE(IJ+NJ)	
1183	BP(IJ)=BP(IJ)+AE(IJ)	
1184	GO TO 845	
1185	842 CONTINUE	
1186	PHI(IJ+NJ)=TE(IJ)	
1187	GO TO 845	
1188	843 CONTINUE	
1189	IJ=IJ+NJ	
1190	IJP=IJ+1	
1191	IJM=IJ-1	
1192	FYN1=FY(IJ)	
1193	FYN2=FY(IJM)	
1194	FYS1=1.-FYN1	
1195	FYS2=1.-FYN2	
1196	DXB=X(IJ)-X(IJM)	
1197	DYB=Y(IJ)-Y(IJM)	
1198	DXBP=QTR*(X(IJ)+NJ)-X(IJ)+X(IJM+NJ)-X(IJM)	
1199	DYBP=QTR*(Y(IJ)+NJ)-Y(IJ)+Y(IJM+NJ)-Y(IJM)	
1200	FAC=(DXB+DXBP+DYB+DYBP)/(DXB**2+DYB**2+SMALL)	
1201	DEL=TE(IJ)*(FYS1-FYN2)*TE(IJP)*FYN1-TE(IJM)*FYS2	
1202	PHI(IJ)=TE(IJ+NJ)-DEL*FAC	
1203	IJ=IJ+NJ	
1204	GO TO 845	
1205	844 CONTINUE	
1206	IF (.NOT. LAY2) GEN(IJ)=GENTEE(J)	

Oct 12 1996 16:36	asmod_2d	Page 18
1207	SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)	
1208	845 CONTINUE	
1209	AE(IJ)=0.0	
1210	840 CONTINUE	
1211	C	
1212	C--- OBSTACLE TREATMENT IS IGNORED HERE	
1213	C	
1214	RETURN	
1215	C	
1216	C-----BOUNDARY CONDITIONS FOR DISSIPATION OF KIN. TURB. ENERGY	
1217	C	
1218	900 CONTINUE	
1219	C-----SOUTH BOUNDARY	
1220	DO 910 I=2,NIM	
1221	IJ=IMNJ(I)+2	
1222	GO TO (911,912,913,914) ITBS(I)	
1223	911 CONTINUE	
1224	SU(IJ)=SU(IJ)+AS(IJ)*ED(IJ-1)	
1225	BP(IJ)=BP(IJ)+AS(IJ)	
1226	GO TO 915	
1227	912 CONTINUE	
1228	PHI(IJ-1)=ED(IJ)	
1229	GO TO 915	
1230	913 CONTINUE	
1231	IJ=IJ-1	
1232	IJP=IJ+NJ	
1233	IMJ=IJ-NJ	
1234	FXE1=FX(IJ)	
1235	FXE2=FX(IMJ)	
1236	FXW1=1.-FXE1	
1237	FXW2=1.-FXE2	
1238	DXB=X(IJ)-X(IMJ)	
1239	DYB=Y(IJ)-Y(IMJ)	
1240	DXBP=QTR*(X(IJ+1)-X(IJ)+X(IMJ+1)-X(IMJ))	
1241	DYBP=QTR*(Y(IJ+1)-Y(IJ)+Y(IMJ+1)-Y(IMJ))	
1242	FAC=(DXB+DXBP+DYB+DYBP)/(DXB**2+DYB**2+SMALL)	
1243	DEL=ED(IJ)*(FXW1-FXE2)+ED(IPJ)*FXE1-ED(IMJ)*FXW2	
1244	PHI(IJ)=ED(IJ+1)-DEL*FAC	
1245	IJ=IJ+1	
1246	GOTO 915	
1247	914 CONTINUE	
1248	TE(IJ)=ABS(TE(IJ))	
1249	SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNS(I))*GREAT	
1250	BP(IJ)=GREAT	
1251	915 CONTINUE	
1252	AS(IJ)=0.0	
1253	910 CONTINUE	
1254	C-----NORTH BOUNDARY	
1255	DO 920 I=2,NIM	
1256	IJ=IMNJ(I)+NJM	
1257	GO TO (921,922,923,924) ITBN(I)	
1258	921 CONTINUE	
1259	SU(IJ)=SU(IJ)+AN(IJ)*ED(IJ+1)	
1260	BP(IJ)=BP(IJ)+AN(IJ)	
1261	GO TO 925	
1262	922 CONTINUE	
1263	PHI(IJ+1)=ED(IJ)	
1264	GO TO 925	
1265	923 CONTINUE	
1266	IJ=IJ+1	
1267	IJP=IJ+NJ	
1268	IMJ=IJ-NJ	
1269	FXE1=FX(IJ)	
1270	FXE2=FX(IMJ)	
1271	FXW1=1.-FXE1	
1272	FXW2=1.-FXE2	
1273	DXB=X(IJ)-X(IMJ)	
1274	DYB=Y(IJ)-Y(IMJ)	
1275	DXBP=QTR*(X(IJ-2)-X(IJ)+X(IMJ-2)-X(IMJ))	
1276	DYBP=QTR*(Y(IJ-2)-Y(IJ)+Y(IMJ-2)-Y(IMJ))	
1277	FAC=(DXB+DXBP+DYB+DYBP)/(DXB**2+DYB**2+SMALL)	

Oct 12 1996 16:36	asmod_2d	Page 19
1278	DEL=ED(IJ)*(FXW1-FXE2)*ED(IJP)*FXY1-ED(IJW)*FXW2	
1279	PHI(IJ)=ED(IJ-1)-DEL*FAC	
1280	IJ=IJ-1	
1281	GO TO 925	
1282	924 CONTINUE	
1283	TE(IJ)=ABS(TE(IJ))	
1284	SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNN(I))*GREAT	
1285	BP(IJ)=GREAT	
1286	925 CONTINUE	
1287	AN(IJ)=0.0	
1288	920 CONTINUE	
1289	C-----WEST BOUNDARY	
1290	DO 930 J=2,NJM	
1291	IJ=IMNJ(2)+J	
1292	GO TO (931,932,933,934) JTBW(J)	
1293	931 CONTINUE	
1294	SU(IJ)=SU(IJ)+AW(IJ)*ED(IJ-NJ)	
1295	BP(IJ)=BP(IJ)+AW(IJ)	
1296	GO TO 935	
1297	932 CONTINUE	
1298	PHI(IJ-NJ)=ED(IJ)	
1299	GO TO 935	
1300	933 CONTINUE	
1301	IJ=J	
1302	IJP=IJ+1	
1303	IJM=IJ-1	
1304	FYN1=FY(IJ)	
1305	FYN2=FY(IJM)	
1306	FYS1=1.-FYN1	
1307	FYS2=1.-FYN2	
1308	DYB=X(IJ)-Y(IJM)	
1309	DXB=X(IJ)-X(IJM)	
1310	DXBP=QTR*(X(IJ+NJ)-X(IJ)+X(IJM+NJ)-X(IJM))	
1311	DYBP=QTR*(Y(IJ+NJ)-Y(IJ)+Y(IJM+NJ)-Y(IJM))	
1312	FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)	
1313	DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2	
1314	PHI(IJ)=ED(IJ+NJ)-DEL*FAC	
1315	IJ=IJ+NJ	
1316	GO TO 935	
1317	934 CONTINUE	
1318	TE(IJ)=ABS(TE(IJ))	
1319	SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNN(J))*GREAT	
1320	BP(IJ)=GREAT	
1321	935 CONTINUE	
1322	AW(IJ)=0.0	
1323	930 CONTINUE	
1324	C-----EAST BOUNDARY	
1325	DO 940 J=2,NJM	
1326	IJ=IMNJ(NIM)+J	
1327	GO TO (941,942,943,944) JTBW(J)	
1328	941 CONTINUE	
1329	SU(IJ)=SU(IJ)+AE(IJ)*ED(IJ+NJ)	
1330	BP(IJ)=BP(IJ)+AE(IJ)	
1331	GO TO 945	
1332	942 CONTINUE	
1333	PHI(IJ+NJ)=ED(IJ)	
1334	GO TO 945	
1335	943 CONTINUE	
1336	IJ=IJ+NJ	
1337	IJP=IJ+1	
1338	IJM=IJ-1	
1339	FYN1=FY(IJ)	
1340	FYN2=FY(IJM)	
1341	FYS1=1.-FYN1	
1342	FYS2=1.-FYN2	
1343	DYB=X(IJ)-Y(IJM)	
1344	DXB=X(IJ)-X(IJM)	
1345	DXBP=QTR*(X(IJ+NJ)-X(IJ)+X(IJM+NJ)-X(IJM))	
1346	DYBP=QTR*(Y(IJ+NJ)-Y(IJ)+Y(IJM+NJ)-Y(IJM))	
1347	FAC=(DXB*DXBP+DYB*DYBP)/(DXB**2+DYB**2+SMALL)	
1348	DEL=ED(IJ)*(FYS1-FYN2)+ED(IJP)*FYN1-ED(IJM)*FYS2	

Oct 12 1996 16:36	asmod_2d	Page 20
1349	PHI(IJ)=ED(IJ-NJ)-DEL*FAC	
1350	IJ=IJ-NJ	
1351	GO TO 945	
1352	944 CONTINUE	
1353	TE(IJ)=ABS(TE(IJ))	
1354	SU(IJ)=CMU75*TE(IJ)*SQRT(TE(IJ))/(CAPPA*DNE(J))*GREAT	
1355	BP(IJ)=GREAT	
1356	945 CONTINUE	
1357	AE(IJ)=0.0	
1358	940 CONTINUE	
1359	C	
1360	RETURN	
1361	END	
1362	C	
1363	C-----SUBROUTINE AMODVIS (TE,ED,VIS,DEN,VISCOS,	
1364	ITBS,TITBN,JTBW,JTBW)	
1365	1	
1366	C-----	
1367	C	
1368	INCLUDE 'gridparam.h'	
1369	INCLUDE 'asm.h'	
1370	C	
1371	DIMENSION TE(NXNY), ED(NXNY), VIS(NXNY), DEN(NXNY)	
1372	DIMENSION ITBS(NX), ITBN(NX), JTBW(NY), JTBW(NY)	
1373	C	
1374	DO 10 I=1,NIM+1	
1375	DO 10 J=1,NJM+1	
1376	IJ=IMNJ(I)+J	
1377	VISOLD=VIS(IJ)	
1378	VIS(IJ)=VISCOS	
1379	IF (ED(IJ).GT.SMALL) THEN	
1380	VIS(IJ)=PMU(IJ)*DEN(IJ)*TE(IJ)**2*CMU/ED(IJ)	
1381	+	
1382	ENDIF	
1383	VIS(IJ)=URFVIS*VIS(IJ)+(1.-URFVIS)*VISOLD	
1384	10 CONTINUE	
1385	C	
1386	IF (LAY2) THEN	
1387	DO 20 I=2,NIM	
1388	DO 210 J=2,NJM	
1389	IJ=IMNJ(I)+J	
1390	IF (PMU(IJ).LT.0.95) VIS(IJ)=VIS2(IJ)	
1391	210 CONTINUE	
1392	20 CONTINUE	
1393	END IF	
1394	C	
1395	RETURN	
1396	END	
1397	C	
1398	C-----	
1399	SUBROUTINE ASOLSIP(PHI,IPHI,RESOR)	
1400	C-----	
1401	C	
1402	INCLUDE 'gridparam.h'	
1403	INCLUDE 'asm.h'	
1404	C	
1405	DIMENSION PHI(NXNY), RES(NXNY)	
1406	DIMENSION BS(NXNY), BN(NXNY), BE(NXNY), BW(NXNY)	
1407	C	
1408	DIMENSION FP(NXNY), RESOR(2)	
1409	C	
1410	C	
1411	NJ = NJM + 1	
1412	NI = NIM + 1	
1413	NINJ = NI*NJ	
1414	DO 5 IJ=1,NINJ	
1415	BN(IJ) = 0.	
1416	BE(IJ) = 0.	
1417	RES(IJ) = 1.	
1418	C	
1419	FP(IJ)=PHI(IJ)	

Oct 12 1996 16:36	asmod_2d	Page 22
1491	RETURN	
1492	END	
1493	C	
1494	C.....A GAUSS ELIMINATION SOLVER	
1495	SUBROUTINE SOLV(A,BB,N)	
1496	DIMENSION A(N,N),B(N),C(N),BB(N),X(N),MME(N)	
1497	EP=1.E-19	
1498	DO 10 J=1,N	
1499	MME(J)=J	
1500	DO 20 I=1,N	
1501	Y=0.	
1502	DO 30 J=I,N	
1503	IF(ABS(A(I,J)).LT.ABS(Y)) GOTO 30	
1504	K=J	
1505	Y=A(I,J)	
1506	CONTINUE	
1507	C	
1508	IF(ABS(Y).LT.EP) THEN	
1509	WRITE(*,*)	
1510	WRITE(9,*)	
1511	DO 35 IA=1,N	
1512	WRITE(*,1000) (A(IA,JA),JA=1,N)	
1513	CONTINUE	
1514	PRINT*, 'THERE IS NO CONVERSE MATRIX'	
1515	STOP 2222	
1516	ENDIF	
1517	C	
1518	Y=1./Y	
1519	DO 40 J=1,N	
1520	C(J)=A(J,K)	
1521	A(J,K)=A(J,I)	
1522	A(J,I)=C(J)*Y	
1523	B(J)=A(I,J)*Y	
1524	A(I,J)=A(I,J)*Y	
1525	A(I,I)=Y	
1526	J=MME(I)	
1527	MME(I)=MME(K)	
1528	MME(K)=J	
1529	DO 11 K=1,N	
1530	IF(K.EQ.I) GOTO 11	
1531	DO 12 J=1,N	
1532	IF(J.EQ.I) GOTO 12	
1533	A(K,J)=A(K,J)-B(J)*C(K)	
1534	CONTINUE	
1535	CONTINUE	
1536	CONTINUE	
1537	DO 33 I=1,N	
1538	DO 44 K=1,N	
1539	IF(MME(K).EQ.I) GOTO 55	
1540	CONTINUE	
1541	IF(K.EQ.I) GOTO 33	
1542	DO 66 J=1,N	
1543	W=A(I,J)	
1544	A(I,J)=A(K,J)	
1545	A(K,J)=W	
1546	IW=MME(I)	
1547	MME(I)=MME(K)	
1548	MME(K)=IW	
1549	CONTINUE	
1550	FORMAT(4X,1P5E13.4)	
1551	DO 50 I=1,N	
1552	X(I)=0.	
1553	DO 50 J=1,N	
1554	X(I)=X(I)+A(I,J)*BB(J)	
1555	DO 60 I=1,N	
1556	BB(I)=X(I)	
1557	RETURN	
1558	END	
1559	C-----	
1560	SUBROUTINE AMODIFY(SUASM,SVASM,SWASM,	
1561	& X,Y,FX,FY,ARE,VOL,R,ICAL,AKSI,DEN,	

Oct 12 1996 16:36	asmod_2d	Page 21
1420	C	
1421	5 CONTINUE	
1422	C	
1423	C	
1424		
1425	DO 10 I=2,NIM	
1426	DO 10 J=2,NJM	
1427	IJ=IMNJ(I)+J	
1428	API=1.0/AP(IJ)	
1429	AP(IJ)=1.0	
1430	AE(IJ)=AE(IJ)*API	
1431	AW(IJ)=AW(IJ)*API	
1432	AN(IJ)=AN(IJ)*API	
1433	AS(IJ)=AS(IJ)*API	
1434	SU(IJ)=SU(IJ)*API	
1435	10 CONTINUE	
1436	C	
1437	DO 20 I=2,NIM	
1438	DO 20 J=2,NJM	
1439	IJ=IMNJ(I)+J	
1440	IJM=IJ-1	
1441	IKJ=IJ-NJ	
1442	BW(IJ)=-AW(IJ)/(1.+ALFAKE*BN(IJ-NJ))	
1443	BS(IJ)=-AS(IJ)/(1.+ALFAKE*BE(IJM))	
1444	POM1=ALFAKE*BW(IJ)*BN(IJM)	
1445	POM2=ALFAKE*BS(IJ)*BE(IJM)	
1446	BP(IJ)=AP(IJ)+POM1+POM2-BW(IJ)*BE(IJM)*BN(IJM)	
1447	BN(IJ)=(-AN(IJ)-POM1)/(BP(IJ)+SMALL)	
1448	BE(IJ)=(-AE(IJ)-POM2)/(BP(IJ)+SMALL)	
1449	20 CONTINUE	
1450	C	
1451	DO 100 L=1,NSWPKE(IPHI)	
1452	RESORP=0.	
1453	DO 30 I=2,NIM	
1454	DO 30 J=2,NJM	
1455	IJ=IMNJ(I)+J	
1456	RES(IJ)=AN(IJ)*PHI(IJ+1)+AS(IJ)*PHI(IJ-1)+AE(IJ)*PHI(IJ+NJ)+	
1457	& AW(IJ)*PHI(IJ-NJ)+SU(IJ)-AP(IJ)*PHI(IJ)	
1458	RESORP=RESORP+ABS(RES(IJ))	
1459	RES(IJ)=(RES(IJ)-BS(IJ)*RES(IJ-1)-BW(IJ)*RES(IJ-NJ))/	
1460	& (BP(IJ)+SMALL)	
1461	30 CONTINUE	
1462	C	
1463	IF(L.EQ.1) RESORKE(IPHI)=RESORP	
1464	RSM=SORKE(IPHI)*RESORKE(IPHI)	
1465	DO 40 I=2,NIM	
1466	II=NIM+2-I	
1467	DO 40 J=2,NJM	
1468	IJ=IMNJ(I)+J	
1469	RES(IJ)=RES(IJ)+JJ	
1470	RES(IJ)=RES(IJ)-BN(IJ)*RES(IJ+1)-BE(IJ)*RES(IJ+NJ)	
1471	PHI(IJ)=PHI(IJ)+RES(IJ)	
1472	40 CONTINUE	
1473	IF(RESORP.LE.RSM) RETURN	
1474	IF(RESORP.LE.RSM) GOTO 200	
1475	100 CONTINUE	
1476	C	
1477	IF(RESORP.GE.RSM.AND.L.GE.NSWPKE(IPHI)) WRITE(*,2)	
1478	2 FORMAT(10X,' SOLSIP DID NOT CONVERGE ')	
1479	C	
1480	200 CONTINUE	
1481	AUX1=0.	
1482	AUX2=0.	
1483	DO 50 I=2,NIM	
1484	DO 50 J=2,NJM	
1485	IJ=IMNJ(I)+J	
1486	AUX1=AUX1+ABS(PHI(IJ)-FP(IJ))	
1487	AUX2=AUX2+ABS(FP(IJ))	
1488	50 CONTINUE	
1489	IF(AUX2.LT.1.E-30) AUX2=1.E-30	
1490	RESOR(IPHI)=AUX1/AUX2	

Oct 12 1996 16:36 asmod_2d Page 23

```

1562 & VISCOS,TE,DNS,DNN,DNW,DNE,ITBS,ITBN,ITBW,JTBE,
1563 & U,V,W,ITER)
1564 C-----
1565 INCLUDE 'gridparam.h'
1566 INCLUDE 'asm.h'
1567 C
1568 DIMENSION UTM(NY),VVM(NY),UVW(NY),UVM(NY),VVM(NY),VVM(NY)
1569 C
1570 DIMENSION X(NXNY),Y(NXNY),EX(NXNY),FY(NXNY),R(NXNY)
1571 DIMENSION DNS(NX),DNN(NX),DNW(NY),DNE(NY)
1572 DIMENSION U(NXNY),V(NXNY),W(NXNY),DEN(NXNY),TE(NXNY)
1573 DIMENSION ITBS(NX),ITBN(NX),ITBW(NY),JTBE(NY)
1574 C
1575 IF (ITER.EQ.1) THEN
1576 REWIND 41
1577 READ(41,*)
1578 READ(41,*)
1579 READ(41,*) CD1,CD2,CMU,ELOG,CAPPA
1580 READ(41,*)
1581 READ(41,*) LRE,LAY2
1582 END IF
1583 C
1584 NI = NIM + 1
1585 NJ = NJM + 1
1586 CMU25=SQRT(SQRT(CMU))
1587 C
1588 C-----SOUTH BOUNDARY
1589 DO 600 I=2,NIM
1590 IJ=IMNJ(I)+2
1591 IF(ITBS(I).EQ.4) THEN
1592 DXB=X(IJ-1)-X(IJ-NJ-1)
1593 DYB=Y(IJ-1)-Y(IJ-NJ-1)
1594 ARW=SQRT(DXB**2+DYB**2)
1595 DXB=DXB/ARW
1596 DYB=DYB/ARW
1597 CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
1598 YPLS=DNS(I)*CONST/VISCOS
1599 IF (YPLS.LE.11.63.OR.LAY2) THEN
1600 TCDEF=VISCOS/DNS(I)
1601 ELSE
1602 UPLUS=LOG(ELOG*YPLS)/CAPPA
1603 TCDEF=CONST/UPLUS
1604 ENDIF
1605 VPINT=U(IJ)*DXB+V(IJ)*DYB
1606 VPINT=VPINT+W(IJ)
1607 VPINT=ABS(VPINT-SQRT(U(IJ-1)*U(IJ-1)+
1608 V(IJ-1)*V(IJ-1)+W(IJ-1)*W(IJ-1)))
1609 GENTS(I)=TCDEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNS(I))
1610 C
1611 ENDIF
1612 C-----NORTH BOUNDARY
1613 IJ=IMNJ(I)+NJM
1614 IF (ITBN(I).EQ.4) THEN
1615 DXB=X(IJ)-X(IJ-NJ)
1616 DYB=Y(IJ)-Y(IJ-NJ)
1617 ARW=SQRT(DXB**2+DYB**2)
1618 DXB=DXB/ARW
1619 DYB=DYB/ARW
1620 CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
1621 YPLS=DNN(I)*CONST/VISCOS
1622 IF (YPLS.LE.11.63.OR.LAY2) THEN
1623 TCDEF=VISCOS/DNN(I)
1624 ELSE
1625 UPLUS=LOG(ELOG*YPLS)/CAPPA
1626 TCDEF=CONST/UPLUS
1627 ENDIF
1628 VPINT=U(IJ)*DXB+V(IJ)*DYB
1629 VPINT=VPINT+W(IJ)
1630 VPINT=ABS(VPINT-SQRT(U(IJ+1)*U(IJ+1)+
1631 V(IJ+1)*V(IJ+1)+W(IJ+1)*W(IJ+1)))
1632 GENTN(I)=TCDEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNN(I))
1633

```

Oct 12 1996 16:36 asmod_2d Page 24

```

1633 C
1634 ENDIF
1635 600 CONTINUE
1636 C-----WEST BOUNDARY
1637 DO 620 J=2,NJM
1638 IJ=IMNJ(J)+J
1639 IF (ITBW(J).EQ.4) THEN
1640 DXB=X(IJ-NJ)-X(IJ-NJ-1)
1641 DYB=Y(IJ-NJ)-Y(IJ-NJ-1)
1642 ARW=SQRT(DXB**2+DYB**2)
1643 DXB=DXB/ARW
1644 DYB=DYB/ARW
1645 CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
1646 YPLS=DNW(J)*CONST/VISCOS
1647 IF (YPLS.LE.11.63.OR.LAY2) THEN
1648 TCDEF=VISCOS/DNW(J)
1649 ELSE
1650 UPLUS=LOG(ELOG*YPLS)/CAPPA
1651 TCDEF=CONST/UPLUS
1652 ENDIF
1653 VPINT=U(IJ)*DXB+V(IJ)*DYB
1654 VPINT=VPINT+W(IJ)
1655 VPINT=ABS(VPINT-SQRT(U(IJ-NJ)*U(IJ-NJ)+
1656 V(IJ-NJ)*V(IJ-NJ)+W(IJ-NJ)*W(IJ-NJ)))
1657 GENTW(J)=TCDEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNW(J))
1658 ENDIF
1659 C-----EAST BOUNDARY
1660 IJ=IMNJ(NIM)+J
1661 IF (ITBE(J).EQ.4) THEN
1662 UP=U(IJ)
1663 VP=V(IJ)
1664 WP=W(IJ)
1665 UWall=U(IJ+NJ)
1666 VWall=V(IJ+NJ)
1667 WWall=W(IJ+NJ)
1668 TEPR=SQRT(TE(IJ))
1669 DELN=DNE(J)
1670 RB=HAF*(R(IJ)+R(IJ-1))
1671 DENS=DEN(IJ)
1672 DXB=X(IJ)-X(IJ-1)
1673 DYB=Y(IJ)-Y(IJ-1)
1674 ARW=SQRT(DXB**2+DYB**2)
1675 DXB=DXB/ARW
1676 DYB=DYB/ARW
1677 CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
1678 YPLS=DNE(J)*CONST/VISCOS
1679 IF (YPLS.LE.11.63.OR.LAY2) THEN
1680 TCDEF=VISCOS/DNE(J)
1681 ELSE
1682 UPLUS=LOG(ELOG*YPLS)/CAPPA
1683 TCDEF=CONST/UPLUS
1684 ENDIF
1685 VPINT=U(IJ)*DXB+V(IJ)*DYB
1686 VPINT=VPINT+W(IJ)
1687 VPINT=ABS(VPINT-SQRT(U(IJ+NJ)*U(IJ+NJ)+
1688 V(IJ+NJ)*V(IJ+NJ)+W(IJ+NJ)*W(IJ+NJ)))
1689 GENTEE(J)=TCDEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNE(J))
1690 ENDIF
1691 620 CONTINUE
1692 C
1693 C----- CALCULATE REYNOLD STRESS GRADIENT TERMS
1694 C----- TO ADD IN THE MOMENTUM EQUATIONS.
1695 C
1696 DO 302 J=2,NJM
1697 I=1
1698 IJ=IMNJ(I)+J
1699 UVM(J)=UV(IJ)*R(IJ)*DEN(IJ)
1700 UWM(J)=U2(IJ)*R(IJ)*DEN(IJ)
1701 VVM(J)=V2(IJ)*R(IJ)*DEN(IJ)
1702 UWM(J)=UW(IJ)*R(IJ)*DEN(IJ)
1703 VWM(J)=VW(IJ)*R(IJ)*DEN(IJ)

```

Oct 12 1996 16:36	asmod_2d	Page 26
1775	IJ=IMNJ(I)+J	
1776	IPJ=IJ+NJ	
1777	IMJ=IJ-NJ	
1778	IJB=IJ+1	
1779	IJM=IJ-1	
1780	DYEP=HAF*(Y(IJ)-Y(IJM)+Y(IMJ)-Y(IMJ-1))	
1781	DYEP=HAF*(Y(IJ)-Y(IMJ)+Y(IJM)-Y(IMJ-1))	
1782	DXEP=HAF*(X(IJ)-X(IJM)+X(IMJ)-X(IMJ-1))	
1783	DXEP=HAF*(X(IJ)-X(IJM)+X(IMJ)-X(IMJ-1))	
1784	DYEP=HAF*(Y(IPJ)+Y(IJ)-Y(IPJ-1)-X(IJM))	
1785	DYEP=HAF*(X(IPJ)+X(IJ)-X(IPJ-1)-X(IJM))	
1786	DXKN=HAF*(X(IPJ)+X(IJ)-X(IPJ-NJ)-X(IMJ))	
1787	DPKN=HAF*(Y(IPJ)+Y(IJ)-Y(IPJ-NJ)-Y(IMJ))	
1788	RPE=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IPJ)+R(IPJ-1))	
1789	RPN=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IPJ)+R(IPJ+1))	
1791 C		
1792 C	STRESS GRADIENT SOURCE TERM IN THE U-MOMENTUM EQUATION	
1793	SUASH(IJ)=(DYEP*DU2EW(IJ)-DXKP*DU2NS(IJ))-	
1794	& (DXKP*DUVNS(IJ)-DXEP*DUVEW(IJ))	
1795 C		
1796 C	STRESS GRADIENT SOURCE TERM IN THE V-MOMENTUM EQUATION	
1797	SVASW(IJ)=(DYEP*DUVEW(IJ)-DYKP*DUVNS(IJ))-	
1798	& (DXKP*DU2NS(IJ)-DXEP*DV2EW(IJ))	
1799	IF(AKSI) SVASW(IJ)=SVASW(IJ)+DEN(IJ)*W2(IJ)*VOL(IJ)/RP	
1800 C		
1801 C	STRESS GRADIENT SOURCE TERM IN THE W-MOMENTUM EQUATION	
1802	IF(ICAL(IRW)) THEN	
1803	SWASH(IJ)=(DYEP*DUWEW(IJ)-DYKP*DUWNS(IJ))-	
1804	& (DXKP*DVWNS(IJ)-DXEP*DWWEW(IJ))	
1805	ENDIF	
1806 C		
1807 C	CONTINUE	
1808 500		
1809 C		
1810 C	RETURN	
1811	END	
1812 C		
1813 C		
1814 C		
1815 C		
1816 C	gridparam.h	
1817 C		
1818	PARAMETER (NX=100)	
1819	PARAMETER (NY=50)	
1820	PARAMETER (NXY=NX*NY)	
1821 C		
1822 C		
1823 C		
1824 C	asm.h	
1825 C		
1826	PARAMETER (HAF=0.5, QTR=0.25)	
1827	PARAMETER (SMALL=1.0E-30, GREAT=1.0E+30)	
1828 C		
1829	COMMON /A1/ AS(NXNY), AN(NXNY), AE(NXNY), AW(NXNY),	
1830	1 AP(NXNY), BP(NXNY), SU(NXNY), APV(NXNY), APV(NXNY)	
1831 C		
1832	COMMON /A2/ SORKE(2), NSWPKE(2), URFKE(2), PRTRKE(2)	
1833 C		
1834	COMMON /A3/ GKE, ALFAKE, RESORKE(2), URFVIS	
1835	COMMON /CONSTT/ CMU, CMU75, ELOG, CAPPA,	
1836	1 C1, C2, C1W, C2W, CK, CE, CD1, CD2	
1837	COMMON /A3/ GENTS(NX), GENTN(NX), GENTW(NY), GENTEE(NY), P11(NXNY),	
1838	1 P22(NXNY), P33(NXNY), P12(NXNY), P13(NXNY), P23(NXNY)	
1839	COMMON /A4/ FUNX(NXNY), FUNY(NXNY), FUNXY(NXNY),	
1840	1 GENDY(NXNY), DUDX(NXNY), DUDY(NXNY), DUDY(NXNY),	
1841	2 DUDY(NXNY), DWDX(NXNY), DWDY(NXNY),	
1842	COMMON /STRESS/ U2(NXNY), V2(NXNY), W2(NXNY),	
1843	1 UV(NXNY), VW(NXNY), UW(NXNY)	
1844 C		
1845	COMMON /WALLKE/ FLR1(NXNY), FLR2(NXNY), FMU(NXNY), VIS2(NXNY),	

Oct 12 1996 16:36	asmod_2d	Page 25
1704 302	CONTINUE	
1705 C		
1706	DO 400 I=2, NIM	
1707	IJ=IMNJ(I)+1	
1708	UUN=U2(IJ)+R(IJ)*DEN(IJ)	
1709	UVN=UV(IJ)+R(IJ)*DEN(IJ)	
1710	VVN=V2(IJ)+R(IJ)*DEN(IJ)	
1711	UUN=UW(IJ)+R(IJ)*R(IJ)*DEN(IJ)	
1712	VVN=VW(IJ)+R(IJ)*R(IJ)*DEN(IJ)	
1713	DO 401 J=2, NJM	
1714	IJ=IMNJ(I)+J	
1715	UUS=UUN	
1716	VVS=VVN	
1717	UVS=UVN	
1718	VMS=VMN	
1719		
1720	U2IJE=U2(IJ+NJ)*R(IJ+NJ)*DEN(IJ+NJ)	
1721	U2IJE=U2(IJ)*R(IJ)*DEN(IJ)	
1722	UVIJE=UV(IJ+NJ)*R(IJ+NJ)*DEN(IJ+NJ)	
1723	UVIJE=UV(IJ)*R(IJ)*DEN(IJ)	
1724	V2IJE=V2(IJ+NJ)*R(IJ+NJ)*DEN(IJ+NJ)	
1725	U2IJE=UW(IJ+NJ)*R(IJ+NJ)*R(IJ+NJ)*DEN(IJ+NJ)	
1726	U2IJE=UW(IJ)*R(IJ)*R(IJ)*DEN(IJ)	
1727	V2IJE=VW(IJ+NJ)*R(IJ+NJ)*R(IJ+NJ)*DEN(IJ+NJ)	
1728	V2IJE=VW(IJ)*R(IJ)*R(IJ)*DEN(IJ)	
1729	U2IUN=U2(IJ+1)*R(IJ+1)*DEN(IJ+1)	
1730	V2IUN=V2(IJ+1)*R(IJ+1)*DEN(IJ+1)	
1731	UVIUN=UV(IJ+1)*R(IJ+1)*DEN(IJ+1)	
1732	UUN=UW(IJ+1)*R(IJ+1)*R(IJ+1)*DEN(IJ+1)	
1733	VMN=VM(IJ+1)*R(IJ+1)*R(IJ+1)*DEN(IJ+1)	
1734 C		
1735	UUE=U2IJE*FX(IJ)+U2IJ*(1.-FX(IJ))	
1736	UVE=U2IJE*FX(IJ)+U2IJ*(1.-FX(IJ))	
1737	VVE=V2IJE*FX(IJ)+V2IJ*(1.-FX(IJ))	
1738	UWE=U2IJE*FX(IJ)+U2IJ*(1.-FX(IJ))	
1739	VWE=V2IJE*FX(IJ)+V2IJ*(1.-FX(IJ))	
1740	UUN=U2IUN*FY(IJ)+U2IJ*(1.-FY(IJ))	
1741	VUN=V2IUN*FY(IJ)+U2IJ*(1.-FY(IJ))	
1742	UUN=U2IUN*FY(IJ)+U2IJ*(1.-FY(IJ))	
1743	VMN=U2IUN*FY(IJ)+U2IJ*(1.-FY(IJ))	
1744	VMN=U2IUN*FY(IJ)+U2IJ*(1.-FY(IJ))	
1745 C		
1746	DU2NS(IJ)=UUN-UUS	
1747	DU2EW(IJ)=UUE-UUV	
1748	DVNS(IJ)=UVN-UVS	
1749	DVEW(IJ)=UVE-UVS	
1750	DV2NS(IJ)=VMN-VVS	
1751	DV2EW(IJ)=VVE-VVW	
1752	DUEW(IJ)=UWE-UWW	
1753	DUNNS(IJ)=UWN-UWS	
1754	DVWEW(IJ)=VWE-VWW	
1755	DVWNS(IJ)=VMN-VMS	
1756 C		
1757	UUN(IJ)=UUE	
1758	VUN(IJ)=UVE	
1759	UVN(IJ)=UVE	
1760	UUN(IJ)=UWE	
1761	VUN(IJ)=VWE	
1762 C		
1763	RP=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IPJ)+R(IPJ-1))	
1764	DP2EW(IJ)=DU2EW(IJ)/RP	
1765	DP2NS(IJ)=DU2NS(IJ)/RP	
1766	DPV2EW(IJ)=DV2EW(IJ)/RP	
1767	DPV2NS(IJ)=DV2NS(IJ)/RP	
1768 C		
1769 401	CONTINUE	
1770 400	CONTINUE	
1771 C		
1772 C		
1773	DO 500 I=2, NIM	
1774	DO 500 J=2, NJM	

1846	1	LRE, LAY2
1847		LOGICAL LAY2,LRE,AKSI,RESTART
1848 c		
1849 c		
1850		

CHAPTER 5

2D/Axisymmetric Full Reynolds Stress (RSM) Turbulence Model

Table of Contents

	page
5.1 Introduction	59
5.2 Theory and Model Equations	59
5.3 Boundary Conditions	67
5.4 Numerical Procedure	68
5.4.1 Wall Reflection Treatment	71
5.6 Module Evaluation	72
References	74
Figures	
Appendix D	76

5.1 Introduction

This report describes a self contained FORTRAN source code to compute turbulent quantities using Launder, Reece and Rodi's [1] second order closure, Reynolds stress model. The module deck is designed to interface with a number of flow solvers to analyse incompressible turbulent internal flows. Detailed description of the model used is given with a special emphasis on the coupling of the mean velocity and Reynolds stresses in the discretization procedure of the generalized coordinate system using a co-located finite volume method. The module was interfaced with the REACT flow solver and tested with benchmark flows including the backward-facing step. The module was also successfully interfaced with the MAST code at the University of Alabama at Huntsville (UAH) and independently tested. The Reynolds stress model implemented produced consistently more accurate simulations than the standard $k-\epsilon$ model.

5.2 Theory and Model Equations

The flow is considered planar or axially symmetric, steady with constant fluid properties. Its mean field may be described by a two-dimensional time averaged equations of continuity and momentum, which can be written as;

$$\frac{\partial \rho U}{\partial x} + \frac{1}{r} \frac{\partial \rho r V}{\partial r} = 0 \quad (1)$$

$$\frac{\partial (\rho r U \Phi)}{\partial x} + \frac{\partial (\rho r V \Phi)}{\partial r} = \frac{\partial}{\partial x} (r \mu \frac{\partial \Phi}{\partial x}) + \frac{\partial}{\partial r} (r \mu \frac{\partial \Phi}{\partial r}) + r S_{\Phi} \quad (2)$$

Φ stands for any of the dependent variables, namely, U and V (axial and radial velocities respectively) and rW (radial distance r multiplied by the tangential velocity W). ρ is the fluid density, μ is the laminar viscosity. S_{Φ} is the source term for the variable Φ and is given by;

$$\text{- Axial direction, } \Phi = U \text{ and } S_U = -\frac{\partial P}{\partial x} - \frac{\partial \overline{\rho u^2}}{\partial x} - \frac{1}{r} \frac{\partial \overline{\rho r u v}}{\partial r}$$

$$\text{- Radial direction, } \Phi = V \text{ and } S_V = -\frac{\partial P}{\partial r} + \frac{\rho W^2}{r} - \frac{2\mu V}{r^2} - \frac{1}{r} \frac{\partial \overline{r r v^2}}{\partial r} - \frac{\partial \overline{r u v}}{\partial x} + \frac{\overline{r w^2}}{r}$$

$$\text{- Tangential direction, } \Phi = rW \text{ and } S_W = -2 \frac{\mu}{r} \frac{\partial r W}{\partial r} - \rho \frac{\overline{r u w}}{\partial x} - \rho \frac{\partial \overline{r v w}}{\partial r} - 2 \overline{r v w}.$$

where u , v and w are the fluctuating velocity components in the axial, radial and azimuthal directions respectively.

Turbulence wall effects in the module are represented by Gibson and Launder [2] version of the high Reynolds number stress transport closure of Launder, Reece and Rodi [1]. The stress closure consists essentially of modeled transport equations for the stresses $\overline{u_i u_j}$ and for axisymmetric swirling flow it includes all the six stresses $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, \overline{uv} , \overline{uw} and \overline{vw} .

The set of differential equations governing the transport of Reynolds stresses ($\overline{u_i u_j}$) is obtained from Navier-Stokes equations by multiplying the equations for the fluctuating components (u_i) and (u_j) by (u_j) and (u_i) respectively, then summing these equations and time averaging the results. The resulting Reynolds stress transport equations are then solved using the mean flow equations to obtain the mean and turbulent flow quantities.

The full transport equations for the Reynolds stresses can be written in a compact form using Cartesian tensor representation as;

$$\frac{1}{r} \frac{\partial \rho r U_k \overline{u_i u_j}}{\partial x_k} - \frac{1}{r} \frac{\partial}{\partial x_k} (r C_k \rho \overline{u_k u_l} \frac{k}{\varepsilon} \frac{\partial \overline{u_i u_j}}{\partial x_l}) = P_{ij} + D_{ij} + \Phi_{ij} - \varepsilon_{ij} \quad (3)$$

Where U_k are the mean velocity components in x_k -direction. The right hand side contains the production term P_{ij} given as

$$P_{ij} = -\rho (\overline{u_i u_k} \frac{\partial U_j}{\partial x_k} + \overline{u_j u_k} \frac{\partial U_i}{\partial x_k}) \quad (4)$$

P_{ij} does not require approximations since it is fully represented by turbulent stresses and mean flow gradients.

The dissipation correlation ε_{ij} arise from the fine-scale of the turbulent motion. At high Reynolds numbers these scales are many orders of magnitude smaller than the large energy containing eddies and turbulence energy cascades down along the eddy-size range with little linkage occurring at intermediate scales, to be ultimately dissipated by the smallest eddies which are unaware of the nature of the mean flow and the large scale turbulence. Therefore, the structure of these fine scale motions responsible for viscous dissipation is isotropic and the dissipation tensor ε_{ij} reduces to

$$2 \nu \frac{\partial \overline{u_i u_j}}{\partial x_k \partial x_k} = \frac{2}{3} \varepsilon \delta_{ij} \quad (5)$$

An additional equations for the dissipation ε is required.

D_{ij} represents the Reynolds stress diffusion which does not in general contribute greatly to the balance of transport of $\overline{u_i u_j}$ except in regions of low stress production by mean strain. This term include contributions of fluctuating pressure-velocity correlations ($\overline{p u_i}$ and $\overline{p u_j}$), triple correlations $\overline{u_i u_j u_k}$ and viscous diffusion $\nu \frac{\partial \overline{u_i u_j}}{\partial x_k}$. Daly and Harlow [3] proposed a simple gradient diffusion hypothesis to model the stress diffusion term in the form

$$D_{ij} = C_s \frac{\partial}{\partial x_k} \rho \frac{k}{\varepsilon} \left[\overline{u_k u_l} \frac{\partial \overline{u_i u_j}}{\partial x_l} \right] \quad (6)$$

with constant C_s is taken to be 0.22. Lien and Leschziner [4] simplified the treatment of the diffusion term to allow an appropriate isotropic diffusivity in the form

$$D_{ij} = \frac{\partial}{\partial x_k} \left[\frac{\mu}{\sigma_k} \frac{\partial}{\partial x_k} (\overline{u_i u_j}) \right] \quad (7)$$

where σ_k is a dimensionless constant. Harlow's proposal for the diffusion term is adopted in the present module since it is based on the fundamental conservation equations for the triple correlations, while Lien & Leschziner's form has a weaker basis in this respect.

Φ_{ij} represents the redistribution of turbulence energy among the normal stresses through the interaction of pressure and strain fluctuations. Modeling the pressure-strain term is the most elaborate and involves the solution of the Poisson equation for pressure fluctuations p . The explicit appearance of the pressure in the correlation is eliminated by taking the divergence of the equation for the fluctuating velocity u_i , thus obtaining a Poisson equation for p . Following a volume integration of the resulting equation subject to the assumption of local mean-flow homogeneity results in three contributions to the pressure-strain correlation Φ_{ij} . One involving just fluctuating quantities $\Phi_{ij,1}$ another arising from the presence of the mean rate of strain $\Phi_{ij,2}$. and a third arising from the surface integral representing wall effects $\Phi_{ij,w}$. Since the primary role of Φ_{ij} is to guide turbulence towards isotropy, Rotta [5] proposed for $\Phi_{ij,1}$;

$$\Phi_{ij,1} = -2 \rho C_1 \varepsilon b_{ij} \quad (8)$$

where $b_{ij} = (\overline{u_i u_j} - \frac{2}{3} \delta_{ij} k) / 2k$ is the dimensionless anisotropy parameter. C_1 is a constant and k and ε are turbulent kinetic energy and energy dissipation respectively. More elaborate models have been proposed such as Lumley [6] and Fu [7] using a nonlinear expression for $\Phi_{ij,1}$. The term $\Phi_{ij,2}$ has been the subject of more extensive research. The traditional linear approach similar to Rotta's work simplifies this correlation to:

$$\Phi_{ij,2} = -C_2 (P_{ij} - \frac{2}{3} \delta_{ij} P) \quad (9)$$

where P is the production of turbulent kinetic energy. Analogous to $\Phi_{ij,1}$ the correlation, $\Phi_{ij,2}$ represents the isotropization of turbulence production tensor with C_2 as a constant. More elaborate models such as that of Speziale, Sarkar and Gatski [8] is based on dynamical systems approach and invariancy concepts. Nonlinear models for $\Phi_{ij,2}$ based on the realizability constraints have been developed, e.g, Shih and Lumley [9] and Fu, Launder and Tselepidakis [10]. The simplified correlations in equations (8) and (9) are used in the present module.

The correlation $\Phi_{ij,w}$ represents the wall damping effects that counteracts the tendency of $\Phi_{ij,1}$ and $\Phi_{ij,2}$ to isotropise the turbulent structure. Since close to a solid wall turbulence approach a state of intense anisotropy associated with a tendency towards a 2D turbulence. Following Shir [11] and Gibson and Launder [2], $\Phi_{ij,w}$ is modeled as the combination of two separate terms;

$$\Phi_{ij,1w} = C_{1w} \rho \frac{\varepsilon}{k} [\overline{u_k u_m} n_k n_m \delta_{ij} - \frac{3}{2} \overline{u_k u_i} n_k n_j - \frac{3}{2} \overline{u_k u_j} n_k n_i] f(\frac{l}{l_n}) \quad (10)$$

$$\Phi_{ij,2w} = C_{2w} [\Phi_{km,2} n_k n_m \delta_{ij} - \frac{3}{2} \Phi_{ij,2} n_k n_j - \frac{3}{2} \Phi_{jk,2} n_k n_i] f(\frac{l}{l_n}) \quad (11)$$

where l_n is the normal distance from the point in question to the wall and $l (= \frac{k^{3/2}}{\varepsilon})$ is the turbulent

length scale. The following relationship is used for the wall damping function

$$f = \frac{C_m^{75} k^{3/4}}{\kappa \varepsilon} \frac{1}{\langle l_n \rangle} \quad (12)$$

where $\langle l_n \rangle$ is the average distance of the point considered from the surrounding surfaces and n_i is a wall-normal unit vector in the i -direction. The constants C_{1w} and C_{2w} have values of 0.5 and 0.3 respectively.

It will be of some value to list the full Reynolds stress equations for axisymmetric swirling flows. Although, the derivations have been carried out within the constraints of Cartesian coordinates, considerations will be given next to the forms applicable to any general curved coordinate system.

In general the transport equation for the Reynolds stresses ($\overline{u_i u_j}$) can be written as;

$$C_{ij} = D_{ij} + P_{ij} + F_{ij} - \varepsilon_{ij} + R_{ij} \quad (14)$$

where C_{ij} , D_{ij} , P_{ij} , F_{ij} and ε represent convection, diffusion, production, pressure-strain and dissipation terms. The term R_{ij} results from the transformation of the equation from plane to axially symmetric conditions and swirl. In Cartesian coordinates, the above terms are summarized below for each stress component;

• $\overline{u^2}$ - equation

$$\begin{aligned} C_{11} &= \frac{1}{r} \frac{\partial}{\partial x} (\rho r U \overline{u^2}) + \frac{1}{r} \frac{\partial}{\partial r} (\rho r V \overline{u^2}) \\ D_{11} &= \frac{1}{r} \frac{\partial}{\partial x} \left[\rho r C_k \overline{u^2} \frac{k}{\varepsilon} \frac{\partial \overline{u^2}}{\partial x} + \rho r C_k \overline{uv} \frac{k}{\varepsilon} \frac{\partial \overline{u^2}}{\partial r} \right] \\ &\quad + \frac{1}{r} \frac{\partial}{\partial r} \left[\rho r C_k \overline{uv} \frac{k}{\varepsilon} \frac{\partial \overline{u^2}}{\partial x} + \rho r C_k \overline{v^2} \frac{k}{\varepsilon} \frac{\partial \overline{u^2}}{\partial r} \right] \\ P_{11} &= -2r \left(\overline{u^2} \frac{\partial U}{\partial x} + \overline{uv} \frac{\partial U}{\partial r} \right) \\ \Phi_{11} &= -\rho C_1 \frac{\varepsilon}{k} \left(\overline{u^2} - \frac{2}{3} k \right) - C_2 \left(P_{11} - \frac{2}{3} P \right) \\ &\quad + \rho C_{1w} \frac{\varepsilon}{k} \left[-2 \overline{u^2} f_x + \overline{v^2} f_y - \overline{uv} f_{xy} \right] \\ &\quad + C_{2w} \left[2 C_2 \left(P_{11} - \frac{2}{3} P \right) f_x - C_2 \left(P_{22} - \frac{2}{3} P \right) f_y + C_2 P_{12} f_{xy} \right] \\ \varepsilon_{11} &= -\frac{2}{3} \rho \varepsilon \end{aligned}$$

• $\overline{v^2}$ - equation;

$$C_{22} = \frac{1}{r} \frac{\partial}{\partial x} (\rho r U \overline{v^2}) + \frac{1}{r} \frac{\partial}{\partial r} (\rho r V \overline{v^2})$$

$$D_{22} = \frac{1}{r} \frac{\partial}{\partial x} \left[\rho r C_k \frac{k}{\varepsilon} (\overline{u^2} \frac{\partial \overline{v^2}}{\partial x} + \overline{uv} \frac{\partial \overline{v^2}}{\partial r}) \right] + \frac{1}{r} \frac{\partial}{\partial r} \left[\rho r C_k \frac{k}{\varepsilon} (\overline{uv} \frac{\partial \overline{v^2}}{\partial x} + \overline{v^2} \frac{\partial \overline{v^2}}{\partial y}) \right]$$

$$P_{22} = -2 \rho \left(\overline{uv} \frac{\partial V}{\partial x} + \overline{v^2} \frac{\partial V}{\partial r} - \overline{vw} \frac{W}{r} \right)$$

$$\Phi_{22} = -C_1 \rho \frac{\varepsilon}{k} (\overline{v^2} - \frac{2}{3} k) - C_2 (P_{22} - \frac{2}{3} P)$$

$$+ \rho C_{1w} \frac{\varepsilon}{k} (\overline{u^2} f_x - 2 \overline{v^2} f_y - \overline{uv} f_{xy})$$

$$+ C_{2w} [-C_2 (P_{11} - \frac{2}{3} P) f_x + 2 C_2 (P_{22} - \frac{2}{3} P) f_y + C_2 P_{12} f_{xy}]$$

$$R_{22} = 2 C_k \rho \frac{k}{\varepsilon} \frac{(\overline{w^2})^2}{r^2} - \frac{2}{r} \frac{\partial}{\partial r} \left(\rho C_k \frac{k}{\varepsilon} (\overline{vw})^2 \right) - 2 \frac{\partial}{\partial x} \left(\rho C_k \frac{k}{\varepsilon} \overline{uw} \frac{\overline{vw}}{r} \right)$$

$$- 2 \rho C_k \frac{k}{\varepsilon} \frac{\overline{w^2}}{r^2} \overline{v^2} - 2 \rho C_k \frac{k}{\varepsilon} \frac{\overline{uw}}{r} \frac{\partial \overline{vw}}{\partial x} - 2 \rho C_k \frac{k}{\varepsilon} \frac{\overline{vw}}{r} \frac{\partial \overline{vw}}{\partial r} + 2 \rho \overline{vw} \frac{W}{r}$$

• $\overline{w^2}$ - equation

$$C_{33} = \frac{1}{r} \frac{\partial}{\partial x} (\rho r U \overline{w^2}) + \frac{1}{r} \frac{\partial}{\partial r} (\rho r V \overline{w^2})$$

$$D_{33} = \frac{1}{r} \frac{\partial}{\partial x} \left[\rho r C_k \frac{k}{\varepsilon} (\overline{u^2} \frac{\partial \overline{w^2}}{\partial x} + \overline{uv} \frac{\partial \overline{w^2}}{\partial r}) \right] + \frac{1}{r} \frac{\partial}{\partial r} \left[\rho r C_k \frac{k}{\varepsilon} (\overline{uv} \frac{\partial \overline{w^2}}{\partial x} + \overline{v^2} \frac{\partial \overline{w^2}}{\partial r}) \right]$$

$$P_{33} = -2 \rho \left(\overline{uw} \frac{\partial W}{\partial x} + \overline{vw} \frac{\partial W}{\partial r} + \overline{w^2} \frac{V}{r} \right)$$

$$F_{33} = -\rho C_1 \frac{\varepsilon}{k} (\overline{w^2} - \frac{2}{3} k) - C_2 (P_{33} - \frac{2}{3} P)$$

$$+ \rho C_{1w} \frac{\varepsilon}{k} [\overline{u^2} f_x + \overline{v^2} f_y + 2 \overline{uv} f_{xy}]$$

$$- C_2 C_{2w} [(P_{11} - \frac{2}{3} P) f_x + (P_{22} - \frac{2}{3} P) f_y - 2 C_2 P_{12} f_{xy}]$$

$$R_{33} = 2 \rho C_k \frac{k}{\varepsilon} \overline{w^2} \frac{\overline{v^2}}{r^2} + \frac{2}{r} \frac{\partial}{\partial r} \left(\rho C_k \frac{k}{\varepsilon} (\overline{vw})^2 \right) + 2 \frac{\partial}{\partial x} \left(\rho C_k \frac{k}{\varepsilon} \overline{uw} \frac{\overline{vw}}{r} \right)$$

$$- 2 \rho C_k \frac{k}{\varepsilon} \frac{(\overline{w^2})^2}{r^2} + 2 \rho C_k \frac{k}{\varepsilon} \frac{\overline{uw}}{r} \frac{\partial \overline{vw}}{\partial x} + 2 \rho C_k \frac{k}{\varepsilon} \frac{1}{r} \overline{vw} \frac{\partial \overline{vw}}{\partial r} - 2 \rho \overline{vw} \frac{W}{r}$$

• \overline{uv} - equation

$$C_{12} = \frac{1}{r} \frac{\partial}{\partial x} (\rho r U \overline{uv}) + \frac{1}{r} \frac{\partial}{\partial r} (\rho r V \overline{uv})$$

$$D_{12} = \frac{1}{r} \frac{\partial}{\partial x} \left[\rho r C_k \frac{k}{\varepsilon} \left(\overline{u^2} \frac{\partial \overline{uv}}{\partial x} + \overline{uv} \frac{\partial \overline{uv}}{\partial r} \right) \right] + \frac{1}{r} \frac{\partial}{\partial r} \left[\rho r C_k \frac{k}{\varepsilon} \left(\overline{uv} \frac{\partial \overline{uv}}{\partial x} + \overline{v^2} \frac{\partial \overline{uv}}{\partial r} \right) \right]$$

$$P_{12} = - \rho \left(\overline{u^2} \frac{\partial V}{\partial x} - \overline{uv} \frac{V}{r} + \overline{v^2} \frac{\partial U}{\partial r} - \overline{uw} \frac{W}{r} \right)$$

$$\Phi_{12} = - \rho C_1 \frac{\varepsilon}{k} \overline{uv} - C_2 P_{12}$$

$$- \frac{3}{2} \rho C_{1w} \frac{\varepsilon}{k} \left[\overline{uv} (f_x + f_y) + (\overline{u^2} + \overline{v^2}) f_{xy} \right] \\ + \frac{3}{2} C_2 C_{2w} \left[(P_{11} + P_{22} - \frac{4}{3} P) f_{xy} + P_{12} (f_x + f_y) \right]$$

$$R_{12} = - \rho C_k \frac{k}{\varepsilon} \overline{w^2} \frac{\overline{uv}}{r^2} - \frac{\partial}{\partial x} \left(\rho C_k \frac{k}{\varepsilon} \frac{(\overline{uw})^2}{r} \right) - \frac{1}{r} \frac{\partial}{\partial r} \left(\rho C_k \frac{k}{\varepsilon} \overline{uw} \overline{vw} \right) \\ - \rho C_k \frac{1}{r} \frac{k}{\varepsilon} \left(\overline{uw} \frac{\partial \overline{uw}}{\partial x} + \overline{vw} \frac{\partial \overline{uw}}{\partial r} \right) + \rho \overline{uw} \frac{W}{r}$$

• \overline{vw} - equation

$$C_{23} = \frac{1}{r} \frac{\partial}{\partial x} (\rho r U \overline{vw}) + \frac{1}{r} \frac{\partial}{\partial r} (\rho r V \overline{vw})$$

$$D_{23} = \frac{1}{r} \frac{\partial}{\partial x} \left[\rho r C_k \frac{k}{\varepsilon} \left(\overline{u^2} \frac{\partial \overline{vw}}{\partial x} + \overline{uv} \frac{\partial \overline{vw}}{\partial r} \right) \right] \\ + \frac{1}{r} \frac{\partial}{\partial r} \left[\rho r C_k \frac{k}{\varepsilon} \left(\overline{uv} \frac{\partial \overline{vw}}{\partial x} + \overline{v^2} \frac{\partial \overline{vw}}{\partial r} \right) \right]$$

$$P_{23} = - \rho \left[\overline{uv} \frac{\partial W}{\partial x} + \overline{uw} \frac{\partial V}{\partial x} + \overline{v^2} \frac{\partial W}{\partial r} + \overline{vw} \frac{\partial V}{\partial r} + \overline{vw} \frac{V}{r} - \overline{w^2} \frac{W}{r} \right]$$

$$\begin{aligned}
\Phi_{23} = & -\rho C_1 \frac{\varepsilon}{k} \overline{vw} - C_2 P_{23} - \frac{3}{2} \rho C_{1w} \frac{\varepsilon}{k} (\overline{uw} f_{xy} + \overline{vw} f_y) \\
& + \frac{3}{2} C_2 C_{2w} (P_{13} f_{xy} + P_{23} f_y) \\
R_{23} = & -\rho (\overline{v^2} - \overline{w^2}) \frac{W}{r} + \rho C_k \frac{k}{\varepsilon} \frac{1}{r} \overline{vw} \frac{\partial}{\partial r} (\overline{v^2} - \overline{w^2}) + \frac{\partial}{\partial x} (\rho C_k \frac{k}{\varepsilon} \overline{uw} \frac{(\overline{v^2} - \overline{w^2})}{r}) \\
& + \frac{1}{r} \frac{\partial}{\partial r} (\rho C_k \frac{k}{\varepsilon} \overline{vw} (\overline{v^2} - \overline{w^2})) - 4 \rho C_k \frac{k}{\varepsilon} \frac{1}{r} \overline{vw} \frac{\overline{w^2}}{r} + \rho C_k \frac{1}{r} \frac{k}{\varepsilon} \overline{uw} \frac{\partial}{\partial x} (\overline{v^2} - \overline{w^2})
\end{aligned}$$

• \overline{uw} -equation

$$\begin{aligned}
C_{13} = & \frac{1}{r} \frac{\partial}{\partial x} (\rho r U \overline{uw}) + \frac{1}{r} \frac{\partial}{\partial r} (\rho r V \overline{uw}) \\
D_{13} = & \frac{1}{r} \frac{\partial}{\partial x} [\rho r C_k \frac{k}{\varepsilon} (\overline{u^2} \frac{\partial \overline{uw}}{\partial x} + \overline{uv} \frac{\partial \overline{uw}}{\partial r})] \\
& + \frac{1}{r} \frac{\partial}{\partial r} [\rho r C_k \frac{k}{\varepsilon} (\overline{uv} \frac{\partial \overline{uw}}{\partial x} + \overline{v^2} \frac{\partial \overline{uw}}{\partial r})] \\
P_{13} = & -\rho (\overline{u^2} \frac{\partial W}{\partial x} + \overline{uv} \frac{\partial W}{\partial r} + \overline{vw} \frac{\partial U}{\partial r} - \overline{uw} \frac{\partial V}{\partial r}) \\
F_{13} = & -\rho C_1 \frac{\varepsilon}{k} \overline{uw} - C_2 P_{13} - \frac{3}{2} \rho C_{1w} \frac{\varepsilon}{k} \overline{uw} f_x \\
& - \frac{3}{2} \rho C_{1w} \frac{\varepsilon}{k} \overline{vw} f_{xy} + \frac{3}{2} C_{2w} C_2 P_{13} f_x + \frac{3}{2} C_2 C_{2w} P_{23} f_{xy} \\
R_{13} = & -\rho \overline{uv} \frac{W}{r} + \rho C_k \frac{k}{\varepsilon} \frac{1}{r} \overline{vw} \frac{\partial \overline{uv}}{\partial r} + \frac{\partial}{\partial x} (\rho C_k \frac{k}{\varepsilon} \overline{uw} \frac{\overline{uv}}{r}) \\
& + \frac{1}{r} \frac{\partial}{\partial r} (\rho C_k \frac{k}{\varepsilon} \overline{vw} \overline{uv}) + \rho C_k \frac{k}{\varepsilon} \frac{\overline{uw}}{r} \frac{\partial \overline{uv}}{\partial x} - \rho C_k \frac{k}{\varepsilon} \frac{\overline{w^2}}{r^2} \overline{uw}
\end{aligned}$$

The turbulence energy dissipation rate ε is determined from its own transport equation;

$$\frac{1}{r} \frac{\partial \rho r U_k \varepsilon}{\partial x_k} = \frac{1}{r} \frac{\partial}{\partial x_k} (r C_{\varepsilon \rho} \frac{k}{\varepsilon} \overline{u_k u_l} \frac{\partial \varepsilon}{\partial x_l}) + C_{\varepsilon I} \frac{\varepsilon}{k} P_k - C_{\varepsilon 2} \rho \frac{\varepsilon^2}{k} \quad (14)$$

where the constants $C_{\varepsilon I}$ and $C_{\varepsilon I}$ have values of 1.44 and 1.92 respectively.

The terms f_x , f_y and f_{xy} appearing in the stress-equation are tied to the orientation of the wall through the wall-damping function f and will be explained later in the wall reflection treatment section.

5.3 Boundary Conditions

To solve the transport equations for the Reynolds stresses, boundary conditions for the stresses are needed. In the present module the log-law based relations are used to bridge the gap between the fully turbulent and viscous near-wall regions. Boundary values for the stresses can be derived by applying the Reynolds stress equations to the near-wall equilibrium flow. It can be shown that the stresses are related to the turbulent kinetic energy $\overline{u_i u_j} = C_{ij} k$, where C_{ij} are constants to be determined. Consider as an example, the log-layer turbulent flow, where $S = \frac{\partial U}{\partial y} = \frac{u_\tau}{\kappa y}$, where u_τ is the friction velocity and κ is Von Karman constant. In the log-layer, the limiting form of the stress equation is obtained by neglecting the convective terms and equating the production to dissipation and setting the wall-distance function $f = 1$, hence the molecular and turbulent diffusion terms can be neglected. Consequently, the normal stress equation for the wall-normal component when simplified with $\Phi_{22} = \rho \frac{2}{3} \varepsilon$ is ;

$$\frac{\overline{v^2}}{k} = \frac{2 (-1 + C_1 + C_2 - 2C_2 C_{2w})}{3 (C_1 + 2C_{1w})} = C_{22} \quad (15)$$

From experimental data, Lien & Leschziner [4] reported a value of $C_{22} \sim 0.249$ for near wall equilibrium turbulence. The most frequently used value of $C_1 = 1.8$ and $C_2 = 0.6$, and from Gibson and Launder [2] $C_{1w} = 0.5$ and $C_{2w} = 0.3$. Substituting these values into equation (13) give a value $C_{22} = 0.247$ which is close to the experimental value. Similarly, these constants also give $C_{11} = 1.09$, $C_{33} = 0.654$ and $C_{12} = -0.255$.

5.4 Numerical Procedure

The conservation equations for the Reynolds stresses and the energy dissipation are integrated over control volumes after transformation of the Cartesian form to body-fitted no-orthogonal coordinates. The equation governing the transport of a scalar property Φ , which stands for the Reynolds stress components and the energy dissipation equation can be written as;

$$\frac{1}{J} \frac{\partial}{\partial \xi^k} [J (\rho U_m \Phi - q_m) \beta_m^k] = S^\Phi \quad (16)$$

where ξ^k represents the curvilinear coordinate frame and J is the Jacobian of the coordinate transformation, and β_m^k represents its cofactors and q_m represents the diffusion flux. Equation (16) is then integrated over discrete control volumes where the dependent variables on the volume faces are approximated by finite-difference representation.

In general the diffusion term is represented as

$$q_m = \Gamma_\Phi \frac{\partial \Phi}{\partial \xi^n} \beta_l^n \quad (17)$$

where Γ_Φ is the diffusion coefficient.

The tensorial form of the diffusivity due to Daly and Harlow [3] is adopted as;

$$\Gamma_\Phi = \rho r C_s \frac{k}{\varepsilon} \overline{u_m u_l} \quad (18)$$

instead of the isotropic diffusivity ($\Gamma_\Phi = \mu_t / \sigma_\Phi$). Utilizing the equilibrium assumption and experimental near-wall stress data, the constant C_s is taken to be 0.22 for the Reynolds stress equations and 0.18 for the turbulent energy dissipation equation. The diffusion term is discretized with a second-order central differencing scheme, while the convective terms are discretized using first or second order upwind differencing scheme.

A special discretization practice for the Reynolds stress gradients is introduced into the finite volume procedure with colocated storage arrangement. This is necessary to avoid the problem of mean velocity-Reynolds stress decoupling that can lead to oscillatory solutions or even divergence of the iterative solution algorithm. The procedure adopted in the present work differs from that of Obi & Peric [12] and that of Lien and Leschziner [4] by accounting for all the driving forces of the Reynolds stresses and not only those given by the gradient-diffusion type process. To illustrate the

origin of the problem, consider the Reynolds stress gradient terms in the axial momentum equation in 2D Cartesian uniform grid for simplicity

$$-\frac{\partial \overline{u^2}}{\partial x} - \frac{\partial \overline{uv}}{\partial y}$$

Now integrating over a control volume surrounding node P (cf. Figure 1) yields;

$$- \int \int \frac{\partial \overline{u^2}}{\partial x} dx dy - \int \int \frac{\partial \overline{uv}}{\partial y} dx dy = - [(\overline{u^2}_e - \overline{u^2}_w) \Delta y_p + (\overline{uv}_n - \overline{uv}_s) \Delta x_p]$$

Now if the cell face values of the $\overline{u^2}_e$, $\overline{u^2}_w$, \overline{uv}_n and \overline{uv}_s are evaluated with linear interpolation, the stress difference expression become

$$- [(\frac{\overline{u^2}_E - \overline{u^2}_W}{2}) \Delta y_P + (\frac{\overline{uv}_N - \overline{uv}_S}{2}) \Delta x_P]$$

and since no P -node shear stress appear in the resulting expression, a checker-board oscillation, similar to that played by the pressure field appear. Therefore, a non-linear interpolation scheme is needed to avoid these odd-even oscillations in the same context of Rhie and Chow [13] for cell face velocities. This means that any cell-face velocity is not merely sensitized to the pressure differences centered on that face but also to the Reynolds stress differences.

Consider the discretized equation for the axial normal stress component $\overline{u^2}$ in general non-orthogonal coordinates;

$$A_P \overline{u^2}_P = \sum_{i=n} A_i \overline{u^2}_i + S_{u^2} \quad (19)$$

where n stands for the cells E, W, N and S neighboring P , A_i are the coefficients for the neighboring cells and S_{u^2} is the source term that includes production, dissipation and pressure-strain redistribution terms as;

$$S_{u^2} = P_{11} - \frac{2}{3} \rho \varepsilon + \Phi_{11}$$

where Φ_{11} combines Rotta's stress isotropization model and isotropization of production model and related wall-correction terms due to Gibson and Launder [2].

$$\begin{aligned} \Phi_{11} = & -\rho C_1 \frac{\varepsilon}{k} (\overline{u^2} - \frac{2}{3} k) - C_2 (P_{11} - \frac{2}{3} P) \\ & + \rho C_{1w} \frac{\varepsilon}{k} (- 2 \overline{u^2} f_x + \overline{v^2} f_y - \overline{uv} f_{xy}) \\ & + 2 C_2 C_{2w} (P_{11} - \frac{2}{3} P) f_x - C_2 C_{2w} (P_{22} - \frac{2}{3} P) f_y + C_2 C_{2w} P_{12} f_{xy} \end{aligned} \quad (20)$$

Rearranging the production terms that contribute to the stress generation and noting that

$P = \frac{1}{2} P_{kk}$, then;

$$S_u^2 = AP_{11} + BP_{22} + CP_{33} + DP_{12} + S_{11} \quad (21)$$

where

$$A = 1 - \frac{2}{3} C_2 + \frac{1}{3} C_2 C_{2w} f_y + \frac{4}{3} C_2 C_{2w} f_x$$

$$B = 1 - \frac{2}{3} C_2 + \frac{1}{3} C_2 C_{2w} f_y + \frac{4}{3} C_2 C_{2w} f_x$$

$$C = \frac{1}{3} C_2 + \frac{1}{3} C_2 C_{2w} (f_y - 2f_x)$$

$$D = C_2 C_{2w} f_{xy}$$

and S_{11} contains the remaining terms.

Substituting for the production terms P_{11} , P_{22} , P_{33} and P_{12} , then equation (19) becomes;

$$\begin{aligned} \overline{u^2}_P = & H_P + 2\rho A [\overline{u^2} (D_1 \Delta U^\xi + D_2 \Delta U^\eta) + \overline{uv} (E_1 \Delta U^\eta + E_2 \Delta U^\xi)]_P \\ & + 2\rho B [\overline{uv} (D_1 \Delta V^\xi + D_2 \Delta V^\eta) + \overline{v^2} (E_1 \Delta V^\xi + E_2 \Delta V^\eta) + \frac{\overline{vw}}{A_P} \frac{W}{r}]_P \\ & + 2\rho C [\overline{uw} (D_1 \Delta W^\xi + D_2 \Delta W^\eta) + \overline{vw} (E_1 \Delta W^\xi + E_2 \Delta W^\eta) + \frac{\overline{w^2}}{A_P} \frac{W}{r}]_P \\ & + \rho D [\overline{u^2} (D_1 \Delta V^\xi + D_2 \Delta V^\eta) + \overline{v^2} (E_1 \Delta U^\eta + E_2 \Delta U^\xi) + \frac{\overline{uv}}{A_P} \frac{V}{r} + \frac{\overline{uw}}{A_P} \frac{W}{r}]_P \\ & + \frac{S_{11}}{A_P} \end{aligned} \quad (22)$$

where

$$H_P = \sum_{i=1}^3 A_i \overline{u^2}_i / A_P$$

$$D_1 = -\Delta y_P^\eta / A_P, D_2 = \Delta y_P^\xi / A_P,$$

$$E_1 = -\Delta x_P^\xi / A_P \text{ and } E_2 = \Delta x_P^\eta / A_P$$

here $\Delta y_P^\eta = (y_n - y_s)$, $\Delta y_P^\xi = (y_e - y_w)$, etc

and $\Delta U^\xi = (U_E - U_P)$, $\Delta V^\xi = (V_E - V_P)$, etc

Now, performing the interpolation practice to obtain east cell-face value of the normal stress ($\overline{u^2}_e$) we obtain;

$$\begin{aligned}
\overline{u^2}_e = & \langle \overline{u^2}_p \rangle - \langle 2\rho A \overline{u^2} D_1 \Delta U \xi \rangle - \langle 2\rho A \overline{uv} E_2 \Delta U \xi \rangle \\
& - \langle 2\rho B \overline{uv} D_1 \Delta V \xi \rangle - \langle 2\rho B \overline{v^2} E_2 \Delta V \xi \rangle \\
& - \langle 2\rho C \overline{uw} D_1 \Delta W \xi \rangle - \langle 2\rho C \overline{vw} E_2 \Delta W \xi \rangle \\
& - \langle \rho D \overline{u^2} D_1 \Delta V \xi \rangle - \langle \rho D \overline{v^2} E_2 \Delta U \xi \rangle \\
& + \langle 2\rho A \overline{u^2} D_1 \rangle \Delta U \xi + \langle 2\rho A \overline{uv} E_2 \rangle \Delta U \xi \\
& + \langle 2\rho B \overline{uv} D_1 \rangle \Delta V \xi + \langle 2\rho B \overline{v^2} E_2 \rangle \Delta V \xi \\
& + \langle 2\rho C \overline{uw} D_1 \rangle \Delta W \xi + \langle 2\rho C \overline{vw} E_2 \rangle \Delta W \xi \\
& + \langle 2\rho D \overline{u^2} D_1 \rangle \Delta V \xi + \langle \rho D \overline{v^2} E_2 \rangle \Delta U \xi
\end{aligned} \tag{23}$$

The brackets \langle and \rangle denote linear interpolation. For instance, on the east face

$$\langle \Phi \rangle = (1-f_\xi) \Phi_P + f_\xi \Phi_E \quad \text{where, } f_\xi = \frac{\Delta x_P}{\Delta x_P + \Delta x_E}$$

Similar expressions can be obtained for $\overline{u^2}_w, \overline{u^2}_n, \overline{u^2}_s, \overline{uv}_w, \overline{uv}_n, \overline{uv}_s$ which are then used to calculate the Reynolds stress gradients in the discretized axial momentum equation. Similarly, expressions for $\overline{uv}_e, \overline{uv}_w, \overline{uv}_n, \overline{uv}_s, \overline{v^2}_e, \overline{v^2}_w, \overline{v^2}_s$ and $\overline{v^2}_n$ can be obtained for the stress gradients in the radial momentum equation and $\overline{uw}_e, \overline{uw}_w, \overline{uw}_n, \overline{uw}_s, \overline{vw}_e, \overline{vw}_w, \overline{vw}_n$ and \overline{vw}_s expressions to evaluate stress gradients in the azimuthal momentum equation.

5.4.1 Wall Reflection Treatment

The wall reflection terms $\Phi_{ij,w}$ appear in the pressure-strain term correlation as wall correction terms ($\Phi_{ij,1w}$ and $\Phi_{ij,2w}$) to counteract the tendency of $\Phi_{ij,1}$ and $\Phi_{ij,2}$ to isotropise the turbulence structure. Special consideration is given to the wall proximity effects on the redistribution process $\Phi_{ij,w}$ with relation to the local orthogonal coordinate system at the wall, cf. figure 2.

At a wall, turbulence approach a state of strong anisotropy associated with the tendency towards a 2D turbulence. The wall-reflection terms ensure that normal stress normal to the wall is not too

high. For body-fitted coordinates, there is a need to consider the tensorial form of the wall reflection terms since they are tied to the orientation of the wall through the damping function term (eq. 12). For a curved surface (figure 2), the wall normal vector $\mathbf{n} = n_1 \mathbf{i}_1 + n_2 \mathbf{i}_2$, where \mathbf{i}_1 and \mathbf{i}_2 are unit vectors in Cartesian coordinates. The Cartesian components of the wall-distance function f are given as;

$$f_x = n_1^2 f, \quad f_y = n_2^2 f \quad \text{and} \quad f_{xy} = n_1 n_2 f$$

where $f_x = n_1^2 (C_\mu^{0.75} k^{1.5} / \kappa \varepsilon) / L_n$ for example and L_n is the normal distance from the wall.

The Reynolds stresses close to the wall are transformed from wall coordinates to Cartesian coordinates by appropriate vector decomposition to give;

$$\begin{aligned} \overline{u^2} &= \widetilde{u^2} t_1^2 + \widetilde{v^2} n_1^2 + 2 \widetilde{uv} t_1 n_1 \\ \overline{v^2} &= \widetilde{u^2} t_2^2 + \widetilde{v^2} n_2^2 + 2 \widetilde{uv} t_2 n_2 \\ \overline{w^2} &= \widetilde{w^2} \\ \overline{uv} &= \widetilde{u^2} t_1 t_2 + \widetilde{v^2} n_1 n_2 + \widetilde{uw} (t_1 n_2 + t_2 n_1) \\ \overline{vw} &= \widetilde{uw} t_2 + \widetilde{vw} n_2 \\ \overline{uw} &= \widetilde{uw} t_1 + \widetilde{vw} n_1 \end{aligned}$$

where $\overline{u^2}$, $\overline{v^2}$ are the Reynolds stresses in Cartesian coordinates and $\widetilde{u^2}$, $\widetilde{v^2}$ are the Reynolds stresses in wall-coordinate, n_1 , n_2 are the Cartesian components of the normal vector component and t_1 , t_2 are the Cartesian components of the tangential vector component.

5.5 Module Evaluation

The RSM module was tested at Rocketdyne after interfacing with the CFD solver REACT and at the University of Alabama at Huntsville (UAH) using own solver (MAST). The first test was on fully developed channel flow with length to height ratio of 50 and a Reynolds number of 2×10^5 based on the channel height. A non-uniform mesh of 101×41 was used with clustering at the walls. Figure 3 shows the fully developed mean velocity profile across the channel. Figure 4 shows the normal Reynolds stress profiles across the channel and figure 5 shows the shear stress profile. Similar results were obtained when the module was interfaced and tested independently at UAH using the MAST code.

The next test problem is that of the backward facing step of Driver and Seegmiller [14]. The calculations were performed using a 101x41 grid points with clustering near the walls. The computational domain had a length of 50H (H is the step height) and a width of 9H. The experimental data were used to specify the inflow conditions for a channel flow calculation where the fully developed profiles at the channel exit were used as the inlet conditions for the backward facing step calculations. Fully developed flow conditions were imposed at the outflow boundary. The boundary conditions for the Reynolds stress equations were arrived at by using the log-law of the wall and assuming local equilibrium conditions close to the wall. It can be shown that the Reynolds stresses are related to the turbulent kinetic energy by;

$$\overline{u_i u_j} = C_{ij} k \quad (24)$$

where C_{ij} is constant. The Reynolds stresses at the vicinity of the wall used are

$$\overline{u^2} = 1.098 k, \quad \overline{v^2} = 0.247 k, \quad \overline{uv} = -0.255 k \quad \text{and} \quad \overline{w^2} = 2k - \overline{u^2} - \overline{v^2} = 0.654 k.$$

Figure 6 shows the stream lines using Launder, Reece and Rodi's model. The computed reattachment length was about 5.8H which is closer to the experimental value of 6.1H than the standard $k-\epsilon$ model (5.35H). The figure also shows a small (turbulence driven) secondary flow region at the base corner of the step which cannot be predicted using the isotropic eddy-viscosity $k-\epsilon$ model. Also, a smaller recirculation region is noted at the top lip of the step which is also driven by turbulence anisotropy (more refined grid may be needed to isolate and study this region). Figure 7, shows the mean velocity profile across the channel at four step heights downstream of the step as compared with the standard $k-\epsilon$ turbulence model predictions. The axial normal turbulent intensity ($\overline{u^2}$) profile across the channel at $x/H=4$ is shown on figure 8 and the radial normal turbulent intensity ($\overline{v^2}$) is shown on figure 9. The shear stress (\overline{uv}) profile across the channel at $x/H=4$ is also shown on figure 10. The results show that the module predicts improved results using the RSM model as compared with the standard $k-\epsilon$ model.

REFERENCES

- [1] Launder, B. E., Reece, G. J. and Rodi, W. "Progress in the Development of a Reynolds Stress Closure" J. Fluid Mech. vol. 68, pp.537- , 1975.
- [2] Gibson, M. and Launder, B. E. "Ground Effects on Pressure Fluctuations in the Atmospheric Boundary Layer" J. Fluid Mech., vol. 86, pt. 3, pp. 491- , 1978.
- [3] Daly, B. and Harlow, F. "Transport Equations in Turbulence" Phys. Fluids, vol. 13, pp.2634- , 1970.
- [4] Lien, F.-S. and Leschziner, M. A. "Second-Moment Modelling of Recirculating Flow With a Non-Orthogonal Finite-Volume Algorithm" Turbulent Shear Flows 8, Springer Verlag, pp. 205-222, 1993.
- [5] Rotta, J. C "Statistische Theorie Nichthomogener Turbulenz" Z. Phys. vol. 129, pp.547- , 1951.
- [6] Lumley, J. L. "Computational Modelling of Turbulent Flows" Adv. Appl. Mech., vol. 18, pp.123- , 1978.
- [7] Fu, S. "Computational Modelling of Turbulent Swirling Flows With Second-Moment Closure" Ph.D. Thesis, University of Manchester, 1988.
- [8] Speziale, C. G, Sarkar, S. and Gatski, T. B. "Modelling the Pressure-Strain Correlation of Turbulence: An Invariant Dynamical Systems Approach" J. Fluid Mech., vol. 227, pp. 245-272, 1991.
- [9] Shih, T. -H. and Lumley, J. L. "Modeling of Pressure Correlation Terms in Reynolds Stress and Scalar Flux Equations" Tec. Rep. FDA-85-3, Cornell University, 1985.
- [10] Fu, S., Launder, B. E. and Tselepidakis, D. P. "Accounting the Effects of High Strain Rates in Modeling the Pressure-Strain Corellation" UMIST Mech.Engng. Dept. Rep. TFD/97/5, 1987.

- [11] Shir, C. C. "A Preliminary Numerical Study of Atmospheric Turbulent Flows in the Idealized Planetary Boundary Layer" J. Atmos. Sci., vol. 30, pp. 1327-1339, 1973.
- [12] Obi, D. and Peric, M. "Second-Moment Calculations Procedure for Turbulent Flows With Collocated Variable Arrangement" AIAA J., vol. 29, no. 4, pp. 585- , 1991.
- [13] Rhie, C. M. and Chow, W. L. "Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation" AIAA J., vol. 21, pp. 1525- , 1983.
- [14] Driver, D. and Seegmiller, H. "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow" AIAA J., vol. 23, pp. 163-171, 1985.

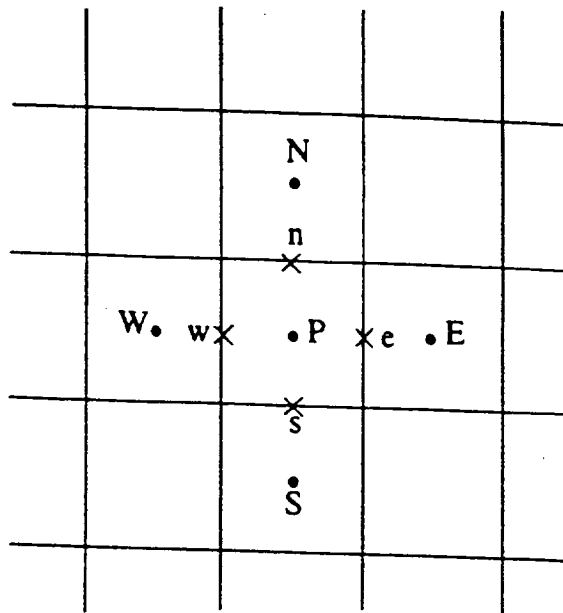


Figure 1. Control volume for node P and it's surroundings

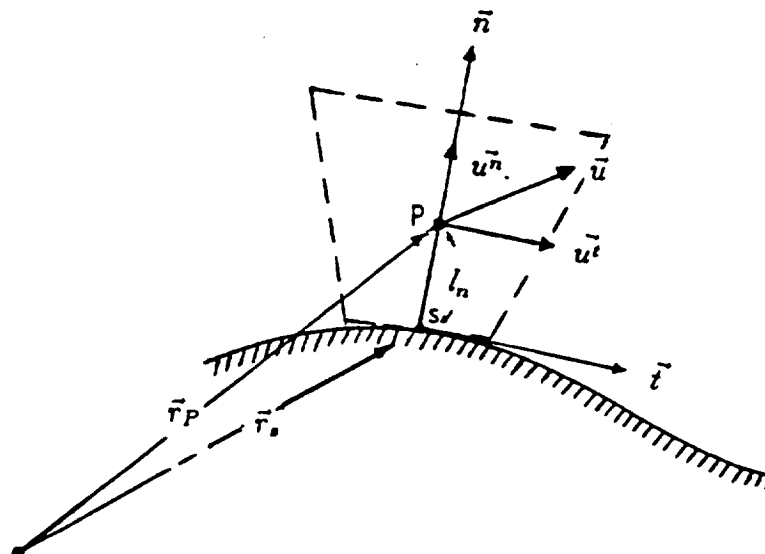


Figure 2. Cartesian and wall-coordinate systems

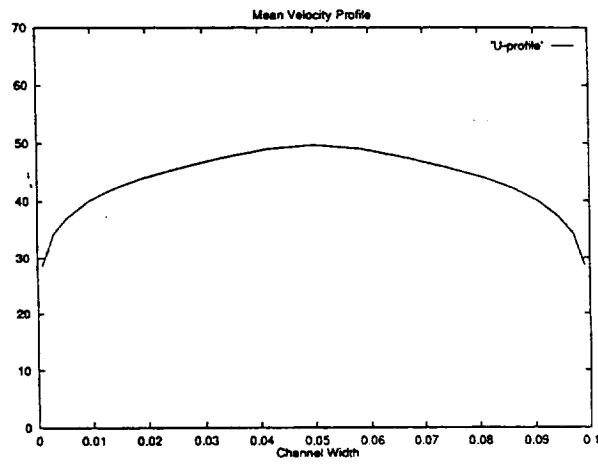


Figure 3. Mean velocity profile across the channel

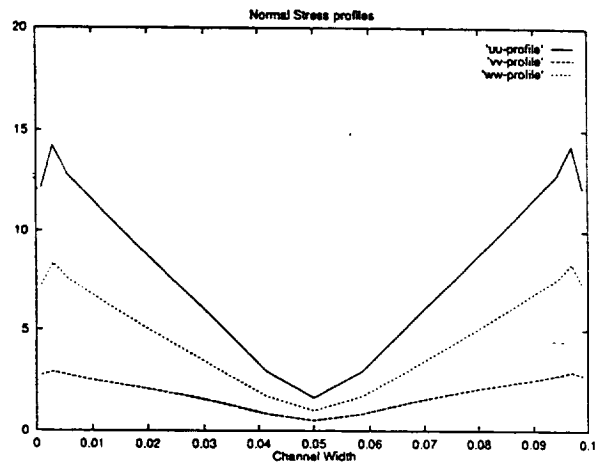


Figure 4. Turbulent intensity profiles across the channel

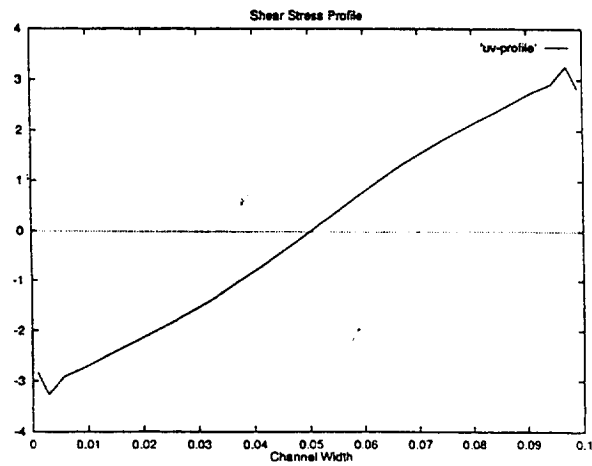


Figure 5. Shear stress profile across the channel

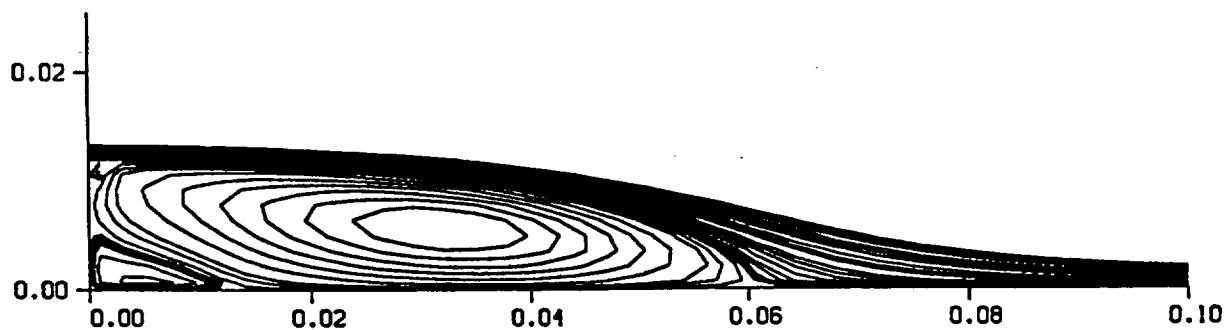


Figure 6. Stream-function contours for the backward facing step flow

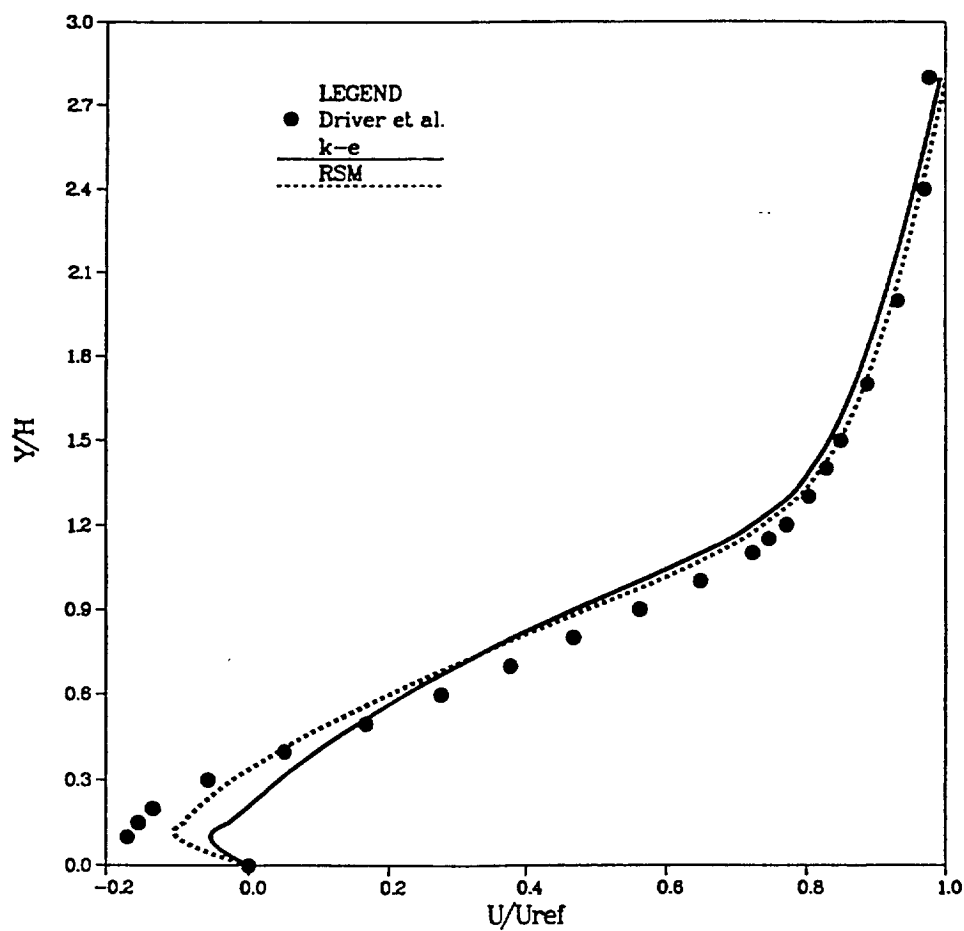


Figure 7. Axial mean velocity profile at $X/H = 4$

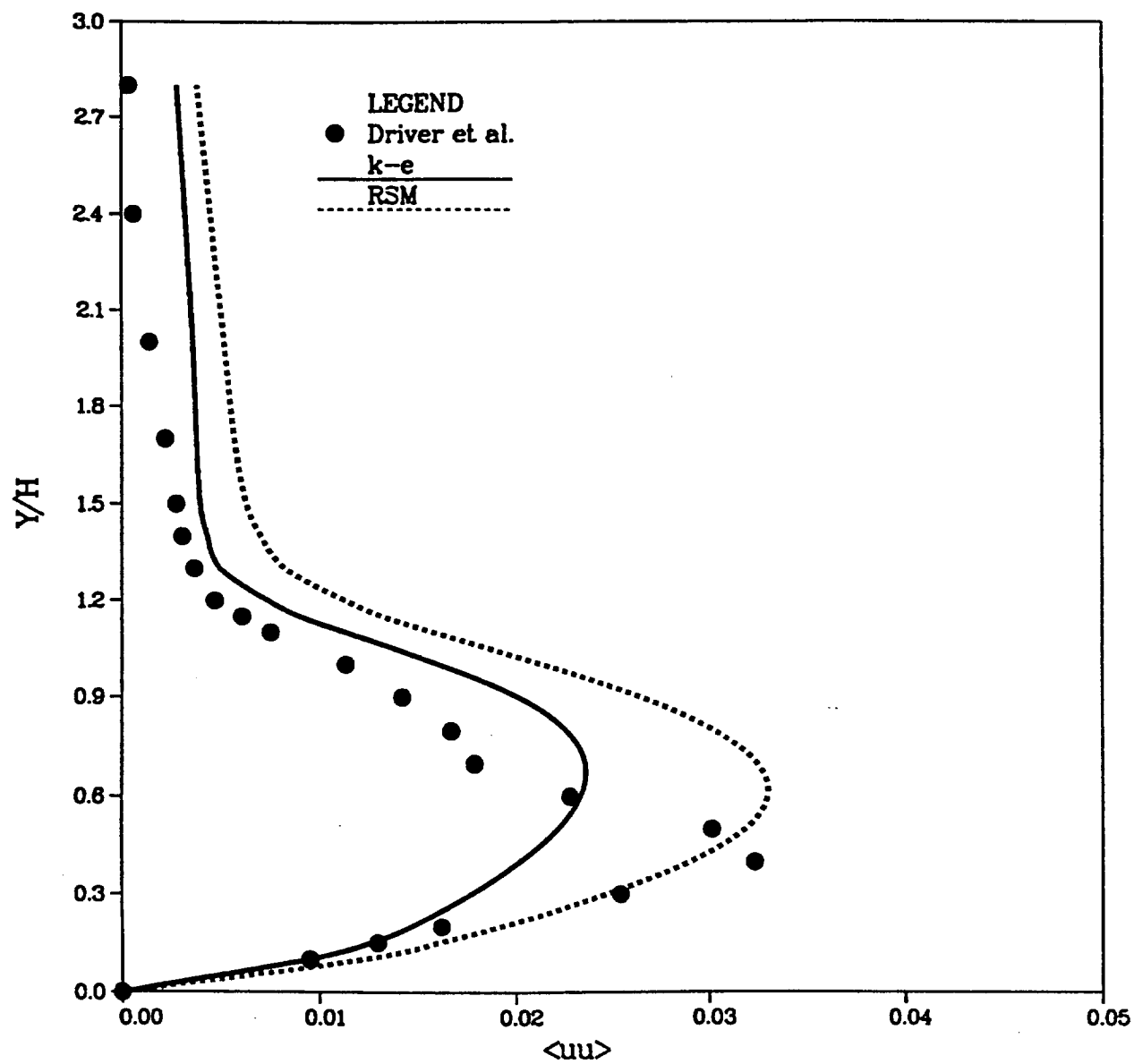


Figure 8. Turbulent intensity \overline{uu} profile
(normalized with U_{ref}^2) at $X/H = 4$

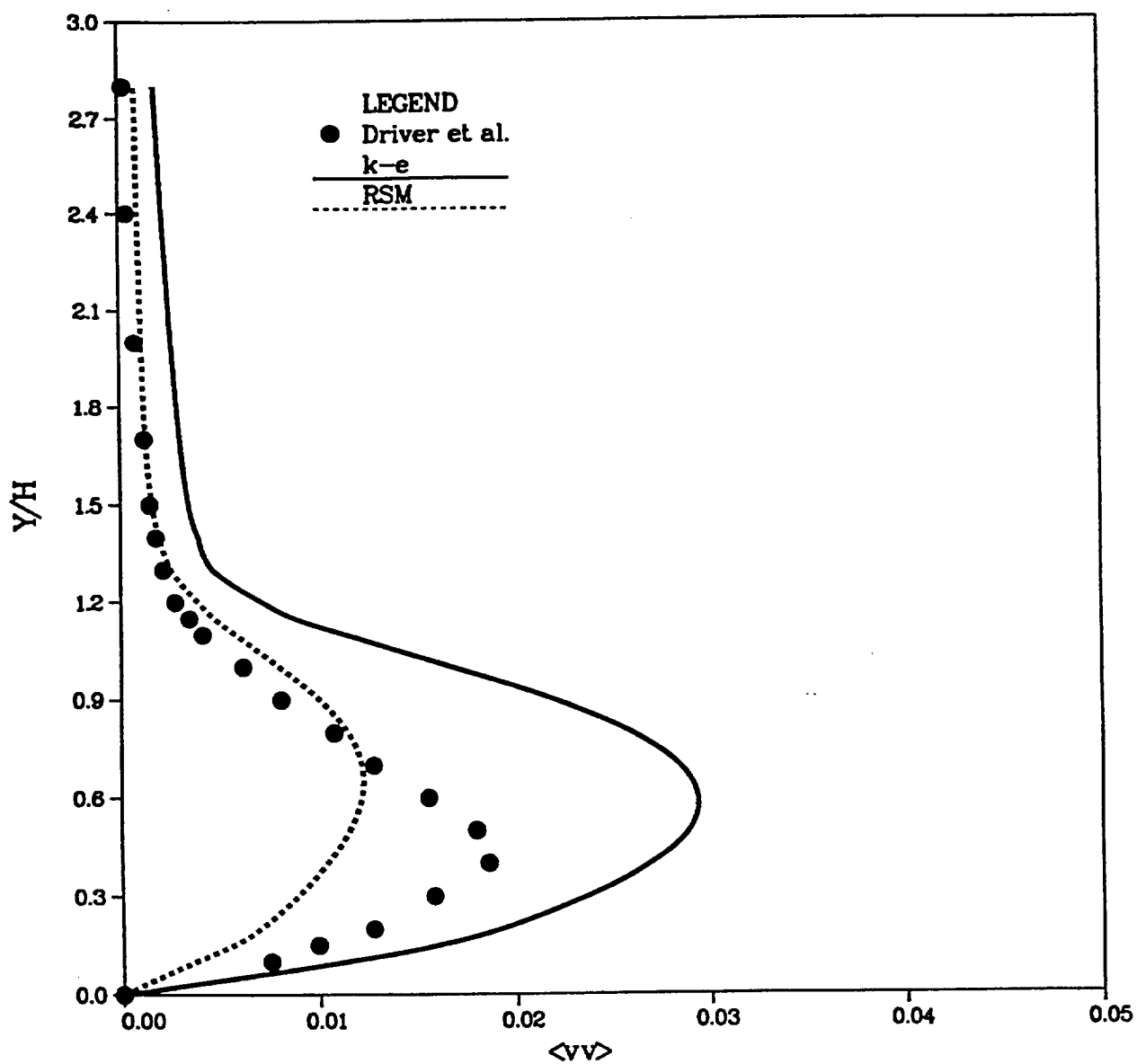


Figure 9. Turbulent stress $\overline{v'v'}$ profile
(normalized with U_{ref}^2) at $X/H = 4$

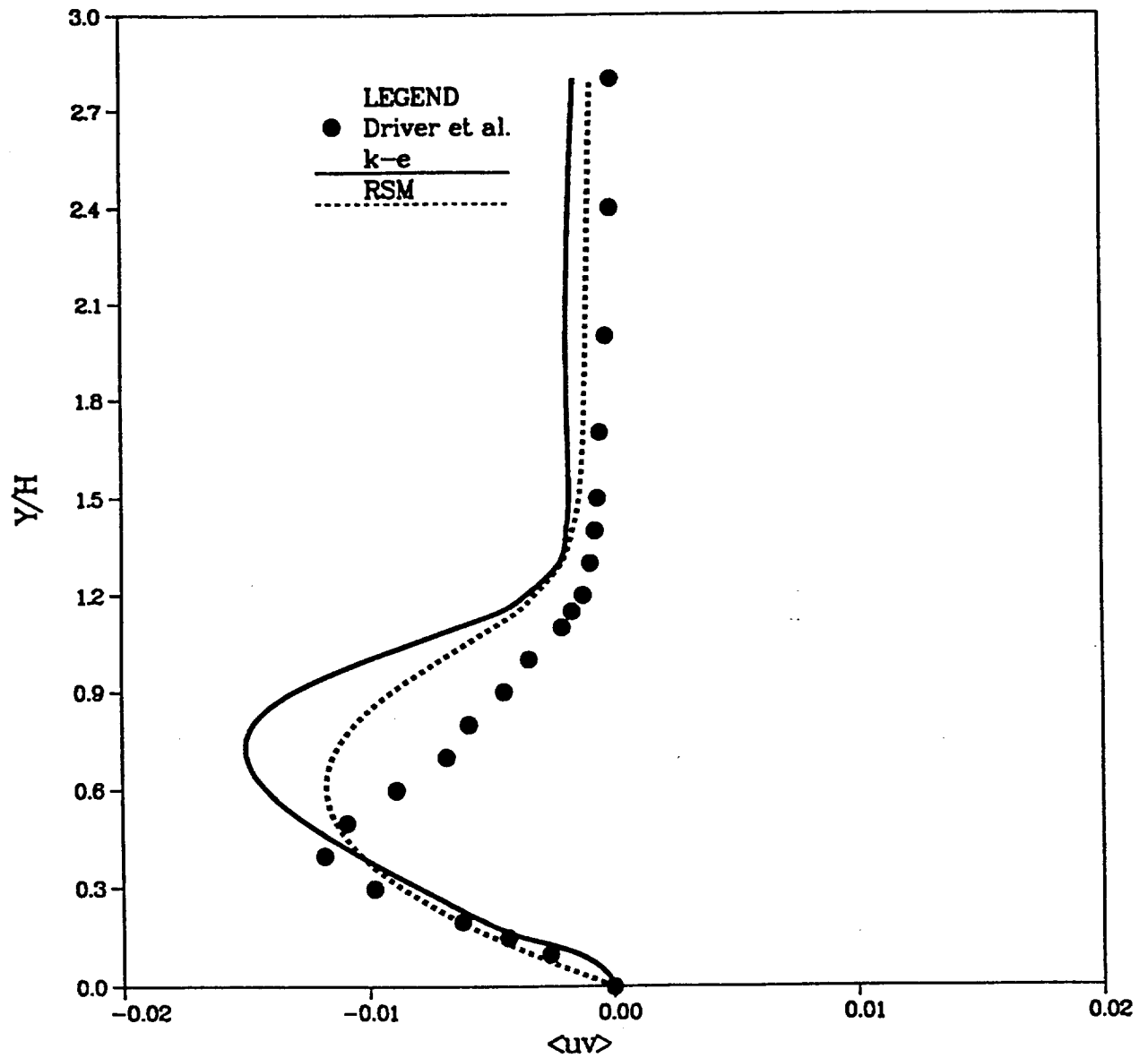


Figure 10. Turbulent shear stress \overline{uv} profile
(normalized with U_{ref}^2) at $X/H = 4$

APPENDIX D

2D/Axisymmetric Reynolds Stress Module Deck

The 2D/axisymmetric Reynolds stress module is a FORTRAN source code to solve 2D/Axisymmetric turbulent flow using the full Reynolds stress model based on Launder, Reece and Rodi[1] when interfaced with a main flow solver. The module consists of the main routine RSMOD that calls a number of subroutines to perform different functions that will be explained below.

3.1 Subroutine RSMOD

This is basically the main routine that reads through its argument list different variables from the calling flow solver which are described below.

List of Argument Variable Names

X	Grid node locations in the x or ξ -direction, dimensioned to X(NX*NY)
Y	Grid node locations in the y or η -direction, dimensioned to Y(NX*NY)
FX	Interpolation factor in the x or ξ -direction.
FY	Interpolation factor in the y or η -direction.
ARE	Control cell areas
VOL	Control cell volumes.
R	Radial distance in the axisymmetric geometry or 1. for planar geometry.
DNS	Normal distance of a cell from the south-boundary dimensioned to NX.
DNN	Normal distance of a cell from the north-boundary dimensioned to NX.
DNE	Normal distance of a cell from the east-boundary dimensioned to NY.
DNW	Normal distance of a cell from the west-boundary dimensioned to NY.
U	Axial or ξ -direction velocity, dimensioned to NX*NY.
V	Radial or ψ -direction velocity, dimensioned to NX*NY.
W	Tangential or azimuthal velocity, dimensioned to NX*NY.
TE	Turbulent kinetic energy, dimensioned to NX*NY.
ED	Turbulent energy dissipation rate, dimensioned to NX*NY.
DEN	Density (assumed constant for incompressible flows).
F1	Mass flux at cell faces in the x or ξ -direction, dimensioned to NX*NY.
F2	Mass flux at cell faces in the y or η -direction, dimensioned to NX*NY.

VISCOUS	Laminar viscosity.
VIS	Eddy viscosity, dimensioned to NX*NY.
RESOR	Residual error for the equations solver, dimensioned to 8.
ITBS	Boundary condition flag along the south boundary dimensioned to NX and must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall e.g., for a wall boundary condition along the south boundary set ITBS to NX*4. Similarly for the other boundaries.
ITBN	Boundary condition flag along the north boundary, dimensioned to NX.
JTBE	Boundary condition flag along the east-boundary dimensioned to NY.
JTBW	Boundary condition flag along the west-boundary dimensioned to NY.
ITER	Iteration number.
ICAL	= 1 for swirl velocity calculations, 0 otherwise.
AKSI	= 1 for axisymmetric flow, 0 otherwise.
RESTART	= 1 if calculations are restarted from a previous run, 0 otherwise.

RSMOD starts by reading the turbulent flow constants, under-relaxation factors and Prandtl/Schmidt numbers for the k and ε equations. These are;

CD1, CD2	constants in the k and ε -equations and are usually set to 1.44 and 1.92 respectively.
CMU, ELOG and CAPPA	also constants in the k and ε -equations and are usually set to 0.09, 9.8 and 0.42 respectively.
GKE	is set to 1 for second-order upwinding of the convective terms in the transport equations.
ALFAKE	is the iteration parameter used in the k and ε -equation solver.
URFVIS	is the underrelaxation factor of the viscosity near the wall.
SORKE(1-8)	are the degree of accuracy for the k , ε , $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, \overline{uv} , \overline{vw} , and \overline{uw} -equations solver respectively.
URFKE(1-8)	are the underrelaxation factors for the k , ε , $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, \overline{uv} , \overline{vw} , and \overline{uw} -equations respectively.
PRTKE(1-8)	are ratio of Prandtl to Schmidt numbers used in the k , ε , $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, \overline{uv} , \overline{vw} , and \overline{uw} -equations respectively.
C ₁ , C ₂	are constants in the RSM model.

C_{1p} and C_{2p}	are the two constants in the wall-reflection terms of the pressure-strain redistribution term.
C_k and C_ε	constants in the diffusion term of the k and ε -equations.
$CUU, CVV, CWW, CUV, CVW, CUW$	are the constants multiplying the kinetic energy for the stress values near the wall.
WREFON	= 1 if the wall reflection terms of the pressure-strain term are to be included, = 0 otherwise.

All variable dimensions considered are one-dimensional. The position of any node is defined as $IJ = (I,J) = (I-1)*NJ + J$, where NI and NJ are the number of grid nodes in the X and Y-directions respectively. It is assumed that grid related data such as cell areas, volumes and interpolation factors be passed to the module from an external grid generator.

Subroutine CALPIJ

This subroutine calculates the production terms of the individual stress components.

Subroutine CALUIUJ

This subroutine solves the transport equations for the turbulent energy ($IPHI=1$), energy dissipation ($IPHI=2$) and Reynolds stresses ($IPHI=3, 4, 5, 6, 7, 8$ for $\overline{u^2}, \overline{v^2}, \overline{w^2}, \overline{uv}, \overline{vw},$ and \overline{uw}). Daly and Harlow [3] gradient stress diffusion form is used in the module instead of the simplified isotropic diffusivity form. This subroutine calls MODUIUJ subroutine that sets the appropriate boundary conditions for the Reynolds stresses. The set of algebraic difference equations are then solved using Stone's strongly implicit solver SOLSIP.

Subroutine MODPIJ

This subroutine modifies the production terms near the wall using the near wall region model.

Subroutine MODUIUJ

This subroutine calculates the near wall boundary conditions for all the variables.

Subroutine SOLSIP

This subroutine solves the system of linear algebraic equations for all the variables using Stone's Implicit Procedure.

Subroutine WALREF

This subroutine calculates the wall reflection terms in the pressure-strain redistribution correlation. It calculates the wall unit normal vectors and the normal distance away from the wall. This is needed to resolve the wall tangential and normal velocity components that are needed to obtain the near-wall values of the Reynolds stresses.

SUBROUTINE WALPARA

This subroutine calculates the normal and tangential wall unit vectors.

[illegible]

```

72 C
73 C --- CALCULATE WALL REFLECTION TERMS
74 C
75 C
76 C
77 C --- IF (WREFON.EQ.1.) CALL WALREF (X,Y,TE,ED,ITBS,ITBN,JTBW,JTBE,JTBE)
78 C
79 C --- CALCULATE TURBULENCE PRODUCTION TERMS
80 C
81 C
82 C
83 C --- CALL CALPIJ (X,Y,FX,FY,ARE,VOL,R,ICAL,AKSI,U,V,W,
84 C & ITBS,ITBN,JTBE,JTBW,ITER)
85 C
86 C --- CALCULATE REYNOLDS STRESSES FROM DIFFERENTIAL EQUATIONS
87 C INCLUDING TURBULENT KINETIC ENERGY AND DISSIPATION RATE
88 C LOOP BELOW FROM 1 TO 8 FOR K, EPS, U2, V2, W2, UV, VW, UW
89 C BUT IPHI FROM 1 TO 12 FOR ALL VARIABLES U, V, W, P, K, EPS,
90 C U2, V2, W2, UV, VW, UW.
91 C
92 C
93 C
94 C
95 C
96 C
97 C
98 C
99 C
100 C
101 C
102 C
103 C
104 C
105 C
106 C --- DO II = 1,8
107 C IPHI = II + 4
108 C DO IJ=1,NINJ
109 C IF(II.EQ.1) PHI(IJ)=TE(IJ)
110 C IF(II.EQ.2) PHI(IJ)=ED(IJ)
111 C IF(II.EQ.3) PHI(IJ)=U2(IJ)
112 C IF(II.EQ.4) PHI(IJ)=V2(IJ)
113 C IF(II.EQ.5) PHI(IJ)=W2(IJ)
114 C IF(II.EQ.6) PHI(IJ)=UV(IJ)
115 C IF(II.EQ.7) PHI(IJ)=VW(IJ)
116 C IF(II.EQ.8) PHI(IJ)=UW(IJ)
117 C ENDDO
118 C IF(ICAL(IPHI)) CALL CALUIUJ(ICAL,IPHI,PHI,AKSI,R,U,V,W,DEN,
119 C & TE,ED,VIS,VISCOS,ITBS,
120 C & ITBN,JTBE,JTBW,WREFON,
121 C & ITER)
122 C ENDDO
123 C
124 C
125 C
126 C --- CHECK TURBULENT ENERGY CALCULATED (TE) WITH BELOW (TERS)
127 C
128 C
129 C
130 C
131 C
132 C
133 C
134 C
135 C
136 C
137 C
138 C
139 C
140 C
141 C
142 C
143 C
144 C
145 C
146 C
147 C
148 C
149 C
150 C
151 C
152 C
153 C
154 C
155 C
156 C
157 C
158 C
159 C
160 C
161 C
162 C
163 C
164 C
165 C
166 C
167 C
168 C
169 C
170 C
171 C
172 C
173 C
174 C
175 C
176 C
177 C
178 C
179 C
180 C
181 C
182 C
183 C
184 C
185 C
186 C
187 C
188 C
189 C
190 C
191 C
192 C
193 C
194 C
195 C
196 C
197 C
198 C
199 C
200 C
201 C
202 C
203 C
204 C
205 C
206 C
207 C
208 C
209 C
210 C
211 C
212 C
213 C
214 C
215 C
216 C
217 C
218 C
219 C
220 C
221 C
222 C
223 C
224 C
225 C
226 C
227 C
228 C
229 C
230 C
231 C
232 C
233 C
234 C
235 C
236 C
237 C
238 C
239 C
240 C
241 C
242 C
243 C
244 C
245 C
246 C
247 C
248 C
249 C
250 C
251 C
252 C
253 C
254 C
255 C
256 C
257 C
258 C
259 C
260 C
261 C
262 C
263 C
264 C
265 C
266 C
267 C
268 C
269 C
270 C
271 C
272 C
273 C
274 C
275 C
276 C
277 C
278 C
279 C
280 C
281 C
282 C
283 C
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
710 C
711 C
712 C
713 C
714 C
715 C
716 C
717 C
718 C
719 C
720 C
721 C
722 C
723 C
724 C
725 C
726 C
727 C
728 C
729 C
730 C
731 C
732 C
733 C
734 C
735 C
736 C
737 C
738 C
739 C
740 C
741 C
742 C
743 C
744 C
745 C
746 C
747 C
748 C
749 C
750 C
751 C
752 C
753 C
754 C
755 C
756 C
757 C
758 C
759 C
760 C
761 C
762 C
763 C
764 C
765 C
766 C
767 C
768 C
769 C
770 C
771 C
772 C
773 C
774 C
775 C
776 C
777 C
778 C
779 C
780 C
781 C
782 C
783 C
784 C
785 C
786 C
787 C
788 C
789 C
790 C
791 C
792 C
793 C
794 C
795 C
796 C
797 C
798 C
799 C
800 C
801 C
802 C
803 C
804 C
805 C
806 C
807 C
80
```

Oct 12 1996 16:39	rsmod_2d	Page 3
143	ENDIF	
144	IF (ITBN(I).EQ.4) THEN	
145	IJ=INNJ(I)+NJ	
146	DXB=X(IJ)-X(IJ-NJ)	
147	DYB=Y(IJ)-Y(IJ-NJ)	
148	FT1N(I)=DXB/PHIP	
149	FT2N(I)=DYB/PHIP	
150	FN1N(I)=DXB/PHIP	
151	FN2N(I)=DYB/PHIP	
152	ENDIF	
153	CONTINUE	
154	10	
155	C	
156	ALONG WEST & EAST BOUNDARIES ---	
157	C	
158	DO 20 J=2,NJM	
159	IF (JTBW(J).EQ.4) THEN	
160	IJ=J	
161	DXB=X(IJ)-X(IJ-1)	
162	DYB=Y(IJ)-Y(IJ-1)	
163	PHIP=SQRT(DXB**2+DYB**2)	
164	FT1W(J)=DXB/PHIP	
165	FT2W(J)=DYB/PHIP	
166	FN1W(J)=DXB/PHIP	
167	FN2W(J)=DYB/PHIP	
168	ENDIF	
169	IF (JTBE(J).EQ.4) THEN	
170	IJ=INNJ(NIM)+J	
171	DXB=X(IJ)-X(IJ-1)	
172	DYB=Y(IJ)-Y(IJ-1)	
173	PHIP=SQRT(DXB**2+DYB**2)	
174	FT1E(J)=DXB/PHIP	
175	FT2E(J)=DYB/PHIP	
176	FN1E(J)=DXB/PHIP	
177	FN2E(J)=DYB/PHIP	
178	ENDIF	
179	20	
180	C	
181	CMU25=SQRT(SQRT(CMU))	
182	CMU75=CMU25**3	
183	FCON=CMU75/CAPPA	
184	C	
185	DO 80 I=2,NJM	
186	DO 80 J=2,NJM	
187	IJ=INNJ(I)+J	
188	RDISNS=0.0	
189	RDISNN=0.0	
190	RDISNE=0.0	
191	RDISNW=0.0	
192	TE(IJ)=ABS(TE(IJ))	
193	COEF=FCON*TE(IJ)**1.5/(ED(IJ)*SMALL)	
194	C---START WITH SOUTH BOUNDARY	
195	IF (ITBS(I).EQ.4) THEN	
196	IJW=INNJ(I)+1	
197	IMJW=IJW-NJ	
198	DXB=X(IJW)-X(IMJW)	
199	DYB=Y(IJW)-Y(IMJW)	
200	XB=HAF*(X(IJW)+X(IMJW))	
201	YB=HAF*(Y(IJW)+Y(IMJW))	
202	YBP=QTR*(X(IJ)-X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
203	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
204	DXBP=XBP-XB	
205	DYBP=YBP-YB	
206	DISNS=DELTA(DXB,DYB,DXBP,DYBP)	
207	DISNS=1.0/(DISNS*SMALL)	
208	ENDIF	
209	C---CHECK NORTH BOUNDARY	
210	IF (ITBN(I).EQ.4) THEN	
211	IJW=INNJ(I)+NJM	
212	IMJW=IJW-NJ	
213	DXB=X(IJW)-X(IMJW)	

Oct 12 1996 16:39	rsmod_2d	Page 4
214	DYB=Y(IJW)-Y(IMJW)	
215	XB=HAF*(X(IJW)+X(IMJW))	
216	YB=HAF*(Y(IJW)+Y(IMJW))	
217	YBP=QTR*(X(IJ)-X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
218	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
219	DXBP=XBP-XB	
220	DYBP=YBP-YB	
221	DISNN=DELTA(DXB,DYB,DXBP,DYBP)	
222	RDISNN=1.0/(DISNN*SMALL)	
223	ENDIF	
224	C---ALONG THE WEST BOUNDARY	
225	IF (JTBW(J).EQ.4) THEN	
226	IJW=J	
227	IMJW=IJW-1	
228	DXB=X(IJW)-X(IMJW)	
229	DYB=Y(IJW)-Y(IMJW)	
230	YB=HAF*(X(IJW)+X(IMJW))	
231	YB=HAF*(Y(IJW)+Y(IMJW))	
232	YBP=QTR*(X(IJ)-X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
233	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
234	DXBP=XBP-XB	
235	DYBP=YBP-YB	
236	DISNW=DELTA(DXB,DYB,DXBP,DYBP)	
237	RDISNW=1.0/(DISNW*SMALL)	
238	ENDIF	
239	C---CHECK EAST BOUNDARY	
240	IF (JTBE(J).EQ.4) THEN	
241	IJW=INNJ(NIM)+J	
242	IMJW=IJW-1	
243	DXB=X(IJW)-X(IMJW)	
244	DYB=Y(IJW)-Y(IMJW)	
245	YB=HAF*(X(IJW)+X(IMJW))	
246	YB=HAF*(Y(IJW)+Y(IMJW))	
247	YBP=QTR*(X(IJ)-X(IJ-1)+X(IJ-NJ-1)+X(IJ-NJ))	
248	YBP=QTR*(Y(IJ)+Y(IJ-1)+Y(IJ-NJ-1)+Y(IJ-NJ))	
249	DXBP=XBP-XB	
250	DYBP=YBP-YB	
251	DISNE=DELTA(DXB,DYB,DXBP,DYBP)	
252	RDISNE=1.0/(DISNE*SMALL)	
253	ENDIF	
254	C	
255	FUNK(IJ)=COEF*(RDISNS*FN1S(I)**2+RDISNN*FN1N(I)**2+RDISNE*FN1E(J)**2+RDISNW*FN1W(J)**2)	
256	FUNK(IJ)=COEF*(RDISNS*FN2S(I)**2+RDISNN*FN2N(I)**2+RDISNE*FN2E(J)**2+RDISNW*FN2W(J)**2)	
257	FUNKY(IJ)=COEF*(RDISNS*FN1S(I)*FN2S(I)+RDISNN*FN1N(I)*FN2N(I)+RDISNE*FN1E(J)*FN2E(J)+RDISNW*FN1W(J)*FN2W(J))	
258	CONTINUE	
259	RETURN	
260	END	
261	C	
262	80	
263	C	
264	C	
265	C	
266	C	
267	C	
268	C	
269	C	
270	SUBROUTINE CALPIJ (X,Y,FX,FY,ARE,VOL,R,ICAL,AKSI,U,V,W,ITBS,ITBN,JTBE,JTBW,ITER)	
271	INCLUDE 'gridparam.h'	
272	INCLUDE 'rsm.h'	
273	DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY)	
274	DIMENSION ARE(NXNY),VOL(NXNY),R(NXNY)	
275	DIMENSION U(NXNY),V(NXNY),W(NXNY)	
276	DIMENSION U1W(NY),U2W(NY),U3W(NY)	
277	DIMENSION ITBS(NX),ITBN(NX),JTBE(NY),JTBW(NY)	
278	DO 10 J=2,NJM	
279	IJ=J	
280	IMJ=IJ-1	
281	IPJ=IJ+NJ	
282		
283		
284		

Oct 12 1996 16:39	rsmosd_2d	Page 5
285	U1W(J)=V(IJ)	
286	U2W(J)=V(IJ)	
287	U3W(J)=W(IJ)	
288 10	CONTINUE	
289 C		
290	DO 101 I=2,NIM	
291	J=1	
292	IJ=IMNJ(I)+J	
293	UN=U(IJ)	
294	VN=V(IJ)	
295	WN=W(IJ)	
296	DO 102 J=2,NJM	
297	IJ=IMNJ(I)+J	
298	IFJ=IJ+NJ	
299	IMJ=IJ-NJ	
300	IJP=IJ+1	
301	IJM=IJ-1	
302	FYE=FX(IJ)	
303	FAW=1.-FXE	
304	FYN=FY(IJ)	
305	FYS=1.-FYN	
306	RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))	
307	DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))	
308	DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))	
309	DYEW=HAF*(YY(IJ)-YY(IMJ)+YY(IJM)-YY(IMJ-1))	
310	DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IMJ)-YY(IMJ-1))	
311 C		
312	US=UN	
313	VS=VN	
314	WS=WN	
315 C		
316	UN=U(IJ)*FYS+U(IJP)*FVN	
317	VN=V(IJ)*FYS+V(IJP)*FVN	
318	WN=W(IJ)*FYS+W(IJP)*FVN	
319 C		
320	UE=U(IJ)*FXW+U(IPJ)*FXE	
321	VE=V(IJ)*FXW+V(IPJ)*FXE	
322	WE=W(IJ)*FXW+W(IPJ)*FXE	
323 C		
324	DUFW=UE-U1W(J)	
325	DUNS=UN-US	
326	DVEW=VE-U2W(J)	
327	DVNS=VN-VS	
328	DWEW=WE-U3W(J)	
329	DNWS=WN-WS	
330 C		
331	DUDX=(DUFW*DYNS-DUNS*DYEW)/ARE(IJ)	
332	DUDY=(DUNS*DXEW-DUEW*DXNS)/ARE(IJ)	
333	DVDX=(DVEW*DYNS-DVNS*DYEW)/ARE(IJ)	
334	DVDY=(DVNS*DXEW-DVEW*DXNS)/ARE(IJ)	
335	DWDY=(DNWS*DYNS-DNWS*DYEW)/ARE(IJ)	
336	DWDY=(DNWS*DXEW-DNWS*DXNS)/ARE(IJ)	
337 C		
338	P11(IJ)=-2.*DEN(IJ)*(U2(IJ)*DUDX+U(IJ)*DUDY)	
339	P22(IJ)=-2.*DEN(IJ)*(U1(IJ)*DVDX+V2(IJ)*DVDY-	
340	& VW(IJ)*W(IJ)/RP)	
341	VDR=0.0	
342	IF(AKSI) VDR=V(IJ)/RP	
343	P33(IJ)=-2.*DEN(IJ)*(UW(IJ)*DWDX+VW(IJ)*DWDY+	
344	& W2(IJ)*VDR)	
345	P12(IJ)=-DEN(IJ)*(U2(IJ)*DVDX+V2(IJ)*DUDY-	
346	& UW(IJ)*W(IJ)/RP-UV(IJ)*VDR)	
347	P13(IJ)=-DEN(IJ)*(U2(IJ)*DWDX+UV(IJ)*DWDY+	
348	& VW(IJ)*DUDY-UW(IJ)*DVDY)	
349	P23(IJ)=-DEN(IJ)*(UV(IJ)*DWDX+UW(IJ)*DVDX+	
350	& V2(IJ)*DWDY+VW(IJ)*DVDY+	
351	& VW(IJ)*VDR-W2(IJ)*W(IJ)/RP)	
352	GEN(IJ)=0.5*(P11(IJ)+P22(IJ)+P33(IJ))	
353 C		
354	U1W(J)=UE	
355	U2W(J)=VE	

Oct 12 1996 16:39	rsmosd_2d	Page 6
356	U3W(J)=WE	
357 C		
358 102	CONTINUE	
359 101	CONTINUE	
360 C		
361 C---	MODIFY GEN-TERMS CLOSE TO A WALL	
362 C		
363	CALL MODPIJ	
364	RETURN	
365	END	
366 C		
367 C		
368 C	SUBROUTINE CALUUIJ(ICAL,IPHI,PHI,AKSI,R,U,V,W,DEN,TE,ED,VIS,	
369	& VISCOS,ITBS,ITBN,JTBW,WREFON,Iter)	
370		
371 C	INCLUDE 'gridparam.h'	
372	INCLUDE 'rsm.h'	
373	DIMENSION PHI(NXNY),FXW(NY),DW(NY)	
374	DIMENSION U(NXNY),V(NXNY),W(NXNY),TE(NXNY),ED(NXNY),	
375	# DEN(NXNY),R(NXNY),A(6,6),VIS(NXNY)	
376	DIMENSION ITBS(NX),ITBN(NX),JTBW(NY)	
377	DIMENSION F222W(NY),F322W(NY),F522W(NY),F622W(NY),	
378	F233W(NY),F333W(NY),F533W(NY),F633W(NY),	
379	F212W(NY),F312W(NY),F412W(NY),F512W(NY),	
380	F123W(NY),F223W(NY),F323W(NY),F523W(NY),	
381	F113W(NY),F213W(NY),F313W(NY),F413W(NY)	
382		
383 C	URFPHI=1./URF(IPHI)	
384		
385 C		
386	IJ=1	
387	PHINE=PHI(IJ)	
388	PHINW(IJ)=PHINE	
389 C		
390	DO 11 J=2,NJM	
391	IJ=J	
392	IJM=IJ-1	
393	IPJ=IJ+NJ	
394	AREE=HAF*(ARE(IJ)+ARE(IPJ))	
395	DXE=XX(IJ)-XX(IJM)	
396	DYE=YY(IJ)-YY(IJM)	
397	DXKS=QTR*(XX(IPJ)+XX(IPJ-1)-XX(IJ)-XX(IJM))	
398	DYKS=QTR*(YY(IPJ)+YY(IPJ-1)-YY(IJ)-YY(IJM))	
399 C		
400	GAMEU2=0.0	
401	GAMEV2=0.0	
402	GAMEUW=0.0	
403	CKK=CK	
404	IF(IPHI.EQ.IED) CKK=CE	
405	IF(ED(IJ).NE.0.0) THEN	
406	TERM=HAF*TEED(IJ)*CKK*DEN(IJ)*(R(IJ)+R(IJM))	
407	GAMEU2=TERM*U2(IJ)/AREE	
408	GAMEV2=TERM*V2(IJ)/AREE	
409	GAMEUW=TERM*UW(IJ)/AREE	
410	ENDIF	
411	DW(IJ)=GAMEU2*DYE**2+GAMEV2*DXE**2	
412	PHISE=PHINE	
413	PHINE=PHI(IJ+1)*FY(IJ)+PHI(IJ)*(1.-FY(IJ))	
414	PHINW(J)=PHINE	
415	SNW(J)=0.0	
416	FXW(J)=1.0	
417	IF(JTBW(J).EQ.3.OR.JTBW(J).EQ.4) GO TO 10	
418	SNW(J)=-((GAMEU2*DYKS+DYE+GAMEV2*DXKS*DXE)*(PHINE-PHISE)+	
419	& 2.*GAMEUW*DXE*DYE*(PHI(IPJ)-PHI(IJ))-	
420	& GAMEUW*(DXKS*DYE+DYKS*DXE)*(PHINE-PHISE))	
421 10	CONTINUE	
422 C		
423 C---	ADD GRADIENT TERMS IN RIJ	
424 C		
425	RP=R(IJ)	
426 C---	V2-EQUATION	

Oct 12 1996 16:39	rsmmod_2d	Page 8
498 C---	GRADIENT TERMS IN RIJ	
499	RP=HAF*(R(IJ)*R(IMJ))	
500 C---	V2-EQUATION	
501	IF(IPHI.EQ.IV2.AND.ICAL(IRW)) THEN	
502	F222N=0.0	
503	F322N=0.0	
504	F522N=0.0	
505	F622N=0.0	
506	IF(RP.GT.0.) THEN	
507	F222N=DEN(IJ)*CK*TEED(IJ)*CK*VW(IJ)**2	
508	F322N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)/RP	
509	F522N=VW(IJ)	
510	F622N=VW(IJ)	
511	ENDIF	
512	ENDIF	
513 C---	W2-EQUATION	
514	IF(IPHI.EQ.IW2.AND.ICAL(IRW)) THEN	
515	F233N=0.0	
516	F333N=0.0	
517	F533N=0.0	
518	F633N=0.0	
519	IF(RP.GT.0.) THEN	
520	F233N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)**2	
521	F333N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)/RP	
522	F533N=VW(IJ)	
523	F633N=VW(IJ)	
524	ENDIF	
525	ENDIF	
526 C---	UV-EQUATION	
527	IF(IPHI.EQ.IUV.AND.ICAL(IRW)) THEN	
528	F212N=0.0	
529	F312N=0.0	
530	F412N=0.0	
531	F512N=0.0	
532	IF(RP.GT.0.) THEN	
533	F212N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)**2/RP	
534	F312N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)/RP	
535	F412N=UV(IJ)	
536	F512N=UV(IJ)	
537	ENDIF	
538	ENDIF	
539 C---	VW-EQUATION	
540	IF(IPHI.EQ.IVW) THEN	
541	F123N=0.0	
542	F223N=0.0	
543	F323N=0.0	
544	F523N=0.0	
545	IF(RP.GT.0.) THEN	
546	F123N=V2(IJ)-W2(IJ)	
547	F223N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*V2(IJ)-W2(IJ)/RP	
548	F323N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*V2(IJ)-W2(IJ)	
549	F523N=V2(IJ)-W2(IJ)	
550	ENDIF	
551	ENDIF	
552 C---	UV-EQUATION	
553	IF(IPHI.EQ.IUV) THEN	
554	F113N=0.0	
555	F213N=0.0	
556	F313N=0.0	
557	F413N=0.0	
558	IF(RP.GT.0.) THEN	
559	F113N=UV(IJ)	
560	F213N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*UV(IJ)/RP	
561	F313N=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*UV(IJ)	
562	F413N=UV(IJ)	
563	ENDIF	
564	ENDIF	
565 C	THE MAIN LOOP - ASSEMBLY OF COEFFICIENTS AND SOURCES	
566 C---		
567 C	DO 101 J=2,NJM	
568		

Oct 12 1996 16:39	rsmmod_2d	Page 7
427	IF(IPHI.EQ.IV2.AND.ICAL(IRW)) THEN	
428	F222W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)**2	
429	F322W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)/RP	
430	F522W(IJ)=VW(IJ)	
431	F622W(IJ)=VW(IJ)	
432	ENDIF	
433 C---	W2-EQUATION	
434	IF(IPHI.EQ.IW2.AND.ICAL(IRW)) THEN	
435	F233W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)**2	
436	F333W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)/RP	
437	F533W(IJ)=VW(IJ)	
438	F633W(IJ)=VW(IJ)	
439	ENDIF	
440 C---	UV-EQUATION	
441	IF(IPHI.EQ.IUV.AND.ICAL(IRW)) THEN	
442	F212W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)**2/RP	
443	F312W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*VW(IJ)	
444	F412W(IJ)=UV(IJ)	
445	F512W(IJ)=UV(IJ)	
446	ENDIF	
447 C---	VW-EQUATION	
448	IF(IPHI.EQ.IVW) THEN	
449	F123W(IJ)=V2(IJ)-W2(IJ)	
450	F223W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*V2(IJ)-W2(IJ)/RP	
451	F323W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*V2(IJ)-W2(IJ)	
452	F523W(IJ)=V2(IJ)-W2(IJ)	
453	ENDIF	
454 C---	UV-EQUATION	
455	IF(IPHI.EQ.IUV) THEN	
456	F113W(IJ)=UV(IJ)	
457	F213W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*UV(IJ)/RP	
458	F313W(IJ)=DEN(IJ)*CK*TEED(IJ)*VW(IJ)*UV(IJ)	
459	F413W(IJ)=UV(IJ)	
460	ENDIF	
461 C	CONTINUE	
462 11	DO 100 I=2,NJM	
464	J=1	
465	IJ=IMNJ(I)+J	
466	IMJ=IJ-NJ	
467	IJP=IJ+1	
468	FXW=1.-FXE	
469	FXE=FX(IJ)	
470	AREN=HAF*(ARE(IJ)+ARE(IJP))	
471	DYN=XX(IJ)-XX(IMJ)	
472	DYN=YY(IJ)-YY(IMJ)	
473	DYET=QTR*(XX(IJP)+XX(IJP-NJ)-XX(IJ)-XX(IMJ))	
474	DYET=QTR*(YY(IJP)+YY(IJP-NJ)-YY(IJ)-YY(IMJ))	
475	GAMNU2=0.0	
476	GAMNV2=0.0	
477	GAMNUV=0.0	
478	CKK=CK	
479	IF(IPHI.EQ.IED) CKK=CE	
480	IF(ED(IJ).NE.0.0) THEN	
481	TERM=HAF*TEED(IJ)*CKK*DEN(IJ)*(R(IJ)+R(IMJ))	
482	GAMNU2=TERM*U2(IJ)/AREN	
483	GAMNV2=TERM*V2(IJ)/AREN	
484	GAMNUV=TERM*UV(IJ)/AREN	
485	ENDIF	
486	ENDIF	
487	DN=GAMNU2*DYN**2+GAMNV2*DXN**2	
488	FVSS=1.0	
489	PHINE=PHI(IJ+NJ)*FXE+PHI(IJ)*FXW	
490	SEWN=0.0	
491	IF(ITBS(I).EQ.3.OR.ITBS(I).EQ.4) GO TO 110	
492	SEWN=-(GAMNU2*DYN*DYET+GAMNV2*DXN*DYET)*(PHINE-PHINW(IJ))+	
493	& 2.*GAMNUV*DXN*DYET*(PHI(IJP)-PHI(IJ))-	
494	& GAMNUV*(DYET*DYN+DXN*DYET)*(PHINE-PHINW(IJ))	
495 110	CONTINUE	
496	PHINW(IJ)=PHINE	
497 C		

Oct 12 1996 16:39	rsmmod_2d	Page 9
569	IJ=IMNJ(I)+J	
570	IPJ=IJ+NI	
571	IMJ=IJ-NJ	
572	IJP=IJ+1	
573	IJM=IJ-1	
574	FXE=FX(IJ)	
575	FXW=1.-FXE	
576	FYN=FY(IJ)	
577	FYS=1.-FYN	
578 C		
579	DYE=XX(IJ)-XX(IJM)	
580	DYE=YY(IJ)-YY(IJM)	
581	DXN=XX(IJ)-XX(IMJ)	
582	DYN=YY(IJ)-YY(IMJ)	
583	DXKS=QTR*(XX(IPJ)-XX(IMJ)+XX(IPJ-1)-XX(IMJ-1))	
584	DYKS=QTR*(YY(IPJ)-YY(IMJ)+YY(IPJ-1)-YY(IMJ-1))	
585	DXET=QTR*(XX(IJP)-XX(IJM)+XX(IJP-NJ)-XX(IJM-NJ))	
586	DYET=QTR*(YY(IJP)-YY(IJM)+YY(IJP-NJ)-YY(IJM-NJ))	
587	AREE=HAF*(ARE(IJ)+ARE(IPJ))	
588	AREN=HAF*(ARE(IJ)+ARE(IJP))	
589 C		
590	CKK=CK	
591	IF(IPHI.EQ.ITE) CKK=CE	
592	GAMEU2=HAF*CKK/AREE*(TEED(IJ)*U2(IJ)*DEN(IJ)*FXW+	
593	& TEED(IPJ)*U2(IPJ)*DEN(IPJ)*FXE)*(R(IJ)+R(IMJ))	
594	& GAMEU2=HAF*CKK/AREE*(TEED(IJ)*V2(IJ)*DEN(IJ)*FXW+	
595	& TEED(IPJ)*V2(IPJ)*DEN(IPJ)*FXE)*(R(IJ)+R(IJM))	
596	& GAMEUV=HAF*CKK/AREE*(TEED(IJ)*UV(IJ)*DEN(IJ)*FXW+	
597	& TEED(IPJ)*UV(IPJ)*DEN(IPJ)*FXE)*(R(IJ)+R(IJM))	
598 C		
599	GAMNU2=HAF*CKK/AREN*(TEED(IJ)*U2(IJ)*DEN(IJ)*FYS+	
600	& TEED(IJP)*U2(IJP)*DEN(IJP)*FYN)*(R(IJ)+R(IMJ))	
601	& GAMNV2=HAF*CKK/AREN*(TEED(IJ)*V2(IJ)*DEN(IJ)*FYS+	
602	& TEED(IJP)*V2(IJP)*DEN(IJP)*FYN)*(R(IJ)+R(IMJ))	
603	& GAMNUV=HAF*CKK/AREN*(TEED(IJ)*UV(IJ)*DEN(IJ)*FYS+	
604	& TEED(IJP)*UV(IJP)*DEN(IJP)*FYN)*(R(IJ)+R(IMJ))	
605 C		
606	DS=DN	
607	DE=GAMEU2*DYE**2+GAMEV2*DXE**2	
608	DN=GAMNU2*DYN**2+GAMNV2*DXN**2	
609 C		
610 C----	LINEAR UPWIND DIFFERENCING	
611 C		
612	AEE=MIN(F1(IJ),0.0)*FX(IPJ)*GRSM	
613	AWW=-MAX(F1(IMJ),0.0)*(1.0-FXW(J))*GRSM	
614	AE1=-MIN(F1(IJ),0.0)*FXE*GRSM	
615	AW1=MAX(F1(IJ),0.0)*(1.0-FX(IMJ))*GRSM	
616	ANN=MIN(F2(IJ),0.0)*FY(IJP)*GRSM	
617	ASS=-MAX(F2(IJM),0.0)*(1.0-FYSS)*GRSM	
618	AN1=-MIN(F2(IJM),0.0)*FYN*GRSM	
619	AS1=MAX(F2(IJ),0.0)*(1.0-FY(IJM))*GRSM	
620 C		
621	AW(IJ)=DW(J)+MAX(F1(IMJ),0.0)-AWW	
622	AE(IJ)=DE-MIN(F1(IJ),0.0)-AEE	
623	AS(IJ)=DS+MAX(F2(IJM),0.0)-ASS	
624	AN(IJ)=DN-MIN(F2(IJ),0.0)-ANN	
625 C		
626	DXKS=QTR*(XX(IPJ)-XX(IMJ)+XX(IPJ-1)-XX(IMJ-1))	
627	DYKS=QTR*(YY(IPJ)-YY(IMJ)+YY(IPJ-1)-YY(IMJ-1))	
628	DXET=QTR*(XX(IJP)-XX(IJM)+XX(IJP-NJ)-XX(IJM-NJ))	
629	DYET=QTR*(YY(IJP)-YY(IJM)+YY(IJP-NJ)-YY(IJM-NJ))	
630 C		
631	PHISE=PHINE	
632	PHINE=(PHI(IJ)+FYS+PHI(IJP)*FYN)*FXW+	
633	& PHINE=(PHI(IPJ)+FYS+PHI(IPJ+1)*FYN)*FXE	
634	SEWS=SEWN	
635	SEWN=-((GAMNU2*DYN+DYET+GAMNV2*DXN+DXET)*	
636	& (PHINE-PHINW(J))-GAMNUV*(DXET+DYN+DXN+DYET)*	
637	& (PHINE-PHINW(J))+2.*GAMNUV*DXN+DYN*	
638	& (PHI(IJP)-PHI(IJ))	
639 C		

Oct 12 1996 16:39	rsmmod_2d	Page 10
640	SNSE=-((GAMEU2*DYKS+DYE+GAMEV2*DXKS+DXE)*	
641	& (PHINE-PHISE)+2.*GAMEUV*DYE+DYE*	
642	& (PHI(IPJ)-PHI(IJ))-GAMEUV*(DXKS+DYE+DYKS+DXE)*	
643	& (PHINE-PHISE))	
644 C		
645	IF(I.EQ.NIM.AND.(JTBE(J).EQ.3.OR.JTBE(J).EQ.4)) SNSE=0.0	
646	IF(J.EQ.NJM.AND.(ITBN(I).EQ.3.OR.ITBN(I).EQ.4)) SEWN=0.0	
647 C		
648	IMJ1=IMJ-NJ	
649 C		
650	IMJ2=MAX(1,IMJ1)	
651	APV(IJ)=SNSE-SNSW(J)+SEWN-SEWS	
652	& ANN*PHI(IJP+1)+ASS*PHI(IJM-1)+	
653	& ANN*PHI(IJP)+AW1*PHI(IMJ)+ANN*PHI(IJP)+	
654	& AE1*PHI(IPJ)+AS1*PHI(IJM)	
655	& AS1*PHI(IJM)	
656	APU(IJ)=AEE+AWW+ANN+ASS+AE1+AW1+AN1+AS1	
657 C		
658 C----	SOURCE TERMS FOR THE KINETIC ENERGY EQUATION ----	
659 C		
660	IF(IPHI.EQ.ITE) THEN	
661	SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)	
662	BP(IJ)=APU(IJ)+VOL(IJ)*ED(IJ)*DEN(IJ)/(TE(IJ)*SMALL)	
663	ENDIF	
664 C		
665 C----	SOURCE TERM OF THE DISSIPATION EQUATION	
666 C		
667	IF(IPHI.EQ.ITE) THEN	
668	SU(IJ)=APV(IJ)+VOL(IJ)*CD1*ED(IJ)*ABS(GEN(IJ))/	
669	& (TE(IJ)+SMALL)	
670	BP(IJ)=APU(IJ)+VOL(IJ)*CD2*DEN(IJ)*ED(IJ)/	
671	& (TE(IJ)+SMALL)	
672	ENDIF	
673 C		
674 C		
675 C----	ADD SOURCE TERMS DUE TO PRODUCTION, DISSIPATION AND	
676 C----	REDISTRIBUTION TERMS IN THE INDIVIDUAL STRESS EQUATIONS	
677 C		
678	TH=2./3.	
679	RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))	
680	RPN=QTR*(R(IJP)+R(IMJ+1)+R(IJ)+R(IMJ))	
681	RPE=QTR*(R(IPJ)+R(IJ)+R(IPJ-1)+R(IJM))	
682	DXEW=HAF*(XX(IJ)-XX(IMJ)+XX(IJM)-XX(IMJ-1))	
683	DXNS=HAF*(XX(IJ)-XX(IJM)+XX(IMJ)-XX(IMJ-1))	
684	DYEW=HAF*(YY(IJ)-YY(IJM)+YY(IJM)-YY(IMJ-1))	
685	DYNS=HAF*(YY(IJ)-YY(IJM)+YY(IJM)-YY(IMJ-1))	
686 C		
687	FEDK=DEN(IJ)*ED(IJ)/(TE(IJ)+SMALL)	
688	FKDE=DEN(IJ)*CK*TEED(IJ)	
689	FKDEN=DEN(IJP)*CK*TEED(IJP)	
690	FKDEE=DEN(IPJ)*CK*TEED(IPJ)	
691 C		
692 C----	U2-EQUATION SOURCES ----	
693 C		
694	IF(IPHI.EQ.IU2) THEN	
695	& SU1=(1.-C2)*P11(IJ)+2./3.*C2*GEN(IJ)+	
696	& C1*DEN(IJ)*ED(IJ)*2./3.	
697	BP1=-((2./3.*DEN(IJ)*ED(IJ)+C1*FEDK*U2(IJ))/	
698	& (U2(IJ)+SMALL)	
699	SU2=2.*C2*C2P*(P11(IJ)-2./3.*GEN(IJ))*FUNX(IJ)+	
700	& C1P*FEDK*(V2(IJ)*FUNY(IJ)-UV(IJ)*FUNX(IJ))-	
701	& C2*C2P*(P22(IJ)-2./3.*GEN(IJ))*FUNY(IJ)-	
702	& P12(IJ)*FUNX(IJ))	
703	BP2=-2.*C1P*FEDK*FUNX(IJ)	
704 C		
705	SU(IJ)=APV(IJ)+(SU1+SU2)*VOL(IJ)	
706	BP(IJ)=APU(IJ)-(BP1+BP2)*VOL(IJ)	
707	ENDIF	
708 C		
709 C----	V2-EQUATION SOURCES ----	
710 C		

Oct 12 1996 16:39	rsmod_2d	Page 11
711	IF (IPHI.EQ.IV2) THEN	
712	F222S=F222N	
713	F322S=F322N	
714	F522S=F522N	
715	F622S=F622N	
716 C		
717	FDUMP=FKDE*VW(IJ)**2	
718	FDMN=FKDEN*VW(IJ)*FYN	
719	F222N=FDUMP*FYS+FDUMN*FYN	
720	FDUMP=FKDE*VW(IJ)*VW(IJ)/RP	
721	FDMN=FKDEN*VW(IJ)*VW(IJ)/RPN	
722	F322N=FDUMP*FYS+FDUMN*FYN	
723	F522N=VW(IJ)*FYS+VW(IJ)*FYN	
724	F622N=VW(IJ)*FYS+VW(IJ)*FYN	
725 C		
726	FDUMP=FKDE*VW(IJ)**2	
727	FDMN=FKDEN*VW(IJ)*FYN	
728	F222E=FDUMP*FXW+FDUMN*FXE	
729	FDMN=FKDE*VW(IJ)*VW(IJ)/RP	
730	FDMN=FKDEN*VW(IJ)*VW(IJ)/RPE	
731	F322E=FDUMP*FXW+FDUMN*FXE	
732	F522E=VW(IJ)*FXW+VW(IJ)*FXE	
733	F622E=VW(IJ)*FXW+VW(IJ)*FXE	
734 C		
735	F2NS=F222N-F222S	
736	F2EW=F222E-F222W(J)	
737	F3NS=F322N-F322S	
738	F3EW=F322E-F322W(J)	
739	F5NS=F522N-F522S	
740	F5EW=F522E-F522W(J)	
741	F6NS=F622N-F622S	
742	F6EW=F622E-F622W(J)	
743 C		
744	DF2DY=(F2NS*DXEW-F2EW*DXNS)/ARE(IJ)	
745	DF3DX=(F3EW*DYNS-F3NS*DYEW)/ARE(IJ)	
746	DF5DX=(F5EW*DYNS-F5NS*DYEW)/ARE(IJ)	
747	DF6DY=(F6NS*DXEW-F6EW*DXNS)/ARE(IJ)	
748 C		
749	SU1=(1.-C2)*P22(IJ)+TH*C2*GEN(IJ)+	
750	& TH*DEN(IJ)*C1*ED(IJ)+2.*DEN(IJ)*VW(IJ)*W(IJ)/RP	
751	BP1=-(TH*DEN(IJ)*ED(IJ)+C1*FEDK*V2(IJ))/(V2(IJ)+SMALL)	
752	SU2=(C1P*FEDK*V2(IJ)-	
753	& C2*C2P*(P11(IJ)-TH*GEN(IJ)))*FUNX(IJ)+	
754	& 2.*C2*C2P*(P22(IJ)-TH*GEN(IJ))*FONY(IJ)+	
755	& (C2*C2P*P12(IJ)-C1P*FEDK*UV(IJ))*FUNKY(IJ)	
756	BP2=-2.*C1P*FEDK*FONY(IJ)	
757 C		
758	TERM1=0.0	
759	TERM2=0.0	
760	TERM3=0.0	
761	TERM5=0.0	
762	TERM6=0.0	
763	BTERM4=0.0	
764	IF (AKSI) THEN	
765	TERM1=2.*FKDE*W2(IJ)**2/RP**2	
766	BTERM4=-2.*FKDE*W2(IJ)/RP**2	
767	IF (ICAL(IRW)) THEN	
768	TERM4=-2.*FKDE*VW(IJ)*DF6DY/RP	
769	TERM5=-2.*FKDE*VW(IJ)*DF5DX/RP	
770	TERM2=-2.*DF2DY/RP	
771	TERM3=-2.*DF3DX	
772	ENDIF	
773	ENDIF	
774	R22=TERM1+TERM2+TERM3+TERM5+TERM6	
775 C		
776	SU(IJ)=APV(IJ)+(SU1+SU2+R22)*VOL(IJ)	
777	BP(IJ)=APU(IJ)-(BP1+BP2+BTERM4)*VOL(IJ)	
778 C		
779	ENDIF	
780 C		
781 C	----- W2-EQUATION SOURCES -----	

Oct 12 1996 16:39	rsmod_2d	Page 12
782	IF (IPHI.EQ.IW2) THEN	
783	F233S=F233N	
784	F333S=F333N	
785	F533S=F533N	
786	F633S=F633N	
787 C		
788	F233N=FKDE*VW(IJ)**2*FYS+FKDEN*VW(IJ)*FYN	
789	F333N=FKDEN*VW(IJ)*VW(IJ)/RP*FYS +	
790	& FKDE*VW(IJ)*VW(IJ)/RPN*FYN	
791	F533N=VW(IJ)*FYS+VW(IJ)*FYN	
792	F633N=VW(IJ)*FYS+VW(IJ)*FYN	
793 C		
794	F233E=FKDE*VW(IJ)**2*FXW+FKDE*VW(IJ)**2*FXE	
795	F333E=FKDE*VW(IJ)*VW(IJ)/RP*FXW +	
796	& FKDE*VW(IJ)*VW(IJ)/RPE*FXE	
797	F533E=VW(IJ)*FXW+VW(IJ)*FXE	
798	F633E=VW(IJ)*FXW+VW(IJ)*FXE	
799 C		
800	F2NS=F233N-F233S	
801	F2EW=F233E-F233W(J)	
802	F3NS=F333N-F333S	
803	F3EW=F333E-F333W(J)	
804	F5NS=F533N-F533S	
805	F5EW=F533E-F533W(J)	
806	F6NS=F633N-F633S	
807	F6EW=F633E-F633W(J)	
808 C		
809	DF2DY=(F2NS*DXEW-F2EW*DXNS)/ARE(IJ)	
810	DF3DX=(F3EW*DYNS-F3NS*DYEW)/ARE(IJ)	
811	DF5DX=(F5EW*DYNS-F5NS*DYEW)/ARE(IJ)	
812	DF6DY=(F6NS*DXEW-F6EW*DXNS)/ARE(IJ)	
813 C		
814	SU1=(1.-C2)*P33(IJ)+TH*C2*GEN(IJ)+	
815	& TH*DEN(IJ)*C1*ED(IJ)-2.*DEN(IJ)*VW(IJ)*W(IJ)/RP	
816	BP1=-(TH*DEN(IJ)*ED(IJ)+FEDK*C1*W2(IJ))/(W2(IJ)+SMALL)	
817	SU2=(C1P*FEDK*V2(IJ)-C2*C2P*	
818	& (P11(IJ)-TH*GEN(IJ)))*FUNX(IJ)+(C1P*FEDK*V2(IJ)-	
819	& C2*C2P*(P22(IJ)-TH*GEN(IJ))*FONY(IJ)+	
820	& (2.*C1P*FEDK*UV(IJ)-2.*C2*C2P*P12(IJ))*FUNKY(IJ)	
821 C		
822	TERM1=0.0	
823	TERM2=0.0	
824	TERM3=0.0	
825	TERM5=0.0	
826	TERM6=0.0	
827	BTERM4=0.0	
828	IF (AKSI) THEN	
829	TERM1=2.*FKDE*W2(IJ)*V2(IJ)/RP**2	
830	BTERM4=-2.*FKDE*W2(IJ)/RP**2	
831	IF (ICAL(IRW)) THEN	
832	TERM2=2.*DF2DY/RP	
833	TERM3=2.*DF3DX	
834	TERM5=2.*FKDE*VW(IJ)*DF5DX/RP	
835	TERM6=2.*FKDE*VW(IJ)*DF6DY/RP	
836	ENDIF	
837	ENDIF	
838	R33=TERM1+TERM2+TERM3+TERM5+TERM6	
839	SU(IJ)=APV(IJ)+(SU1+SU2+R33)*VOL(IJ)	
840	BP(IJ)=APU(IJ)-(BP1+BTERM4)*VOL(IJ)	
841 C		
842	ENDIF	
843 C		
844 C	----- UV-EQUATION SOURCES -----	
845	IF (IPHI.EQ.IUV) THEN	
846	F212S=F212N	
847	F312S=F312N	
848	F412S=F412N	
849	F512S=F512N	
850 C		
851	F212N=FKDE*VW(IJ)**2/RP*FYS+FKDEN*VW(IJ)*FYN	
852	F312N=FKDE*VW(IJ)*VW(IJ)/RP*FYS+FKDEN*VW(IJ)*FYN	

Oct 12 1996 16:39	rsmod_2d	Page 15
995	TERM2=DF2DX	
996	TERM3=DF3DY/RP	
997	BTERR4=FKDE*DF4DX/RP	
998	BTERR5=-FKDE*W2(IJ)/RP**2	
999	TERM6=-DEN(IJ)*UV(IJ)*W(IJ)/RP	
1000	RIJ=TERM1+TERM2+TERM3+TERM6	
1001 C		
1002	SU(IJ)=APV(IJ)*(SU1+RIJ)*VOL(IJ)	
1003	BP(IJ)=APU(IJ)-(BP1+BTERR4+BTERR5)*VOL(IJ)	
1004 C		
1005	ENDIF	
1006 C		
1007	F222W(J)=F222E	
1008	F322W(J)=F322E	
1009	F522W(J)=F522E	
1010	F622W(J)=F622E	
1011 C		
1012	F233W(J)=F233E	
1013	F333W(J)=F333E	
1014	F533W(J)=F533E	
1015	F633W(J)=F633E	
1016 C		
1017	F212W(J)=F212E	
1018	F312W(J)=F312E	
1019	F412W(J)=F412E	
1020	F512W(J)=F512E	
1021 C		
1022	F123W(J)=F123E	
1023	F223W(J)=F223E	
1024	F323W(J)=F323E	
1025	F523W(J)=F523E	
1026 C		
1027	F113W(J)=F113E	
1028	F213W(J)=F213E	
1029	F313W(J)=F313E	
1030	F413W(J)=F413E	
1031 C		
1032	SNW(J)=SNSE	
1033	PHNW(J)=PHINE	
1034	FYS=FY(IJM)	
1035	FYWW(J)=FY(IWJ)	
1036	DW(J)=DE	
1037 C		
1038 101	CONTINUE	
1039 100	CONTINUE	
1040 C		
1041 C-----	PROBLEM MODIFICATION AND BOUNDARY CONDITIONS	
1042 C		
1043	CALL MODIIUJ	
1044 C		
1045	DO 200 I=2,NIM	
1046	DO 201 J=2,NJM	
1047	IJ=IMNJ(I)+J	
1048	IMJ=IJ-NJ	
1049	IJM=IJ-1	
1050	AP(IJ)=AW(IJ)+RE(IJ)+AN(IJ)+AS(IJ)+BP(IJ)	
1051	AP(IJ)=AP(IJ)*URPHI	
1052	SU(IJ)=SU(IJ)+(1.-URF(IPHI))*AP(IJ)*PHI(IJ)	
1053	IF(IPHI.EQ.IV2) APUV(IJ)=1./AP(IJ)	
1054	IF(IPHI.EQ.IV2) APV(IJ)=1./AP(IJ)	
1055	IF(IPHI.EQ.IVW) APUV(IJ)=1./AP(IJ)	
1056	IF(IPHI.EQ.IVW) APVW(IJ)=1./AP(IJ)	
1057	IF(IPHI.EQ.IVW) APW(IJ)=1./AP(IJ)	
1058 201	CONTINUE	
1059 200	CONTINUE	
1060 C		
1061 C-----	SOLVING F. D. EQUATIONS	
1062 C		
1063	CALL SIPSOL(PHI,IPHI)	
1064 C		
1065	RETURN	

Oct 12 1996 16:39	rsmod_2d	Page 16
1066	END	
1067 C		
1068 C		
1069 C		
1070	SUBROUTINE CALDUVW	
1071 C		
1072	IMPLICIT DOUBLE PRECISION (A-H,O-Z)	
1073 C	SUBROUTINE TO CALCULATE MEAN-VELOCITY GRADIENTS AND	
1074 C	---THE REYNOLDS STRESS TERMS AT THE CELL FACES	
1075 C	---TO INCREASE THE COUPLING BETWEEN THE MEAN-VELOCITY AND	
1076 C	---REYNOLDS STRESSES IN THE MOMENTUM EQUATIONS.	
1077 C		
1078	INCLUDE 'zsm.common.block'	
1079	COMMON/DU123/ DU2EW(NXNY), DU2NS(NXNY), DU2NS(NXNY),	
1080	& DU2EW(NXNY), DU2EW(NXNY), DU2EW(NXNY), DU2EW(NXNY),	
1081	& DU2NS(NXNY), DU2NS(NXNY), DU2NS(NXNY), DU2NS(NXNY),	
1082 C		
1083	DIMENSION DU1(NXNY), DU2(NXNY), DV1(NXNY), DV2(NXNY),	
1084	& DW1(NXNY), DW2(NXNY)	
1085	DIMENSION U1W(NY), V1W(NY), W1W(NY)	
1086	DIMENSION U1W(NY), U1W(NY), V1W(NY), V1W(NY), V1W(NY),	
1087 C		
1088	DO 302 J=2,NJM	
1089	I=1	
1090	IJ=IMNJ(I)+J	
1091	U1W(IJ)=U(IJ)	
1092	V1W(IJ)=V(IJ)	
1093	W1W(IJ)=W(IJ)	
1094	U1W(IJ)=UV(IJ)*R(IJ)*DEN(IJ)	
1095	U1W(IJ)=U2(IJ)*R(IJ)*DEN(IJ)	
1096	V1W(IJ)=V2(IJ)*R(IJ)*DEN(IJ)	
1097	U1W(IJ)=UW(IJ)*R(IJ)*DEN(IJ)	
1098	V1W(IJ)=VW(IJ)*R(IJ)*DEN(IJ)	
1099 302	CONTINUE	
1100 C		
1101	DO 300 I=2,NIM	
1102	IJ=IMNJ(I)+1	
1103	U1N=U(IJ)	
1104	V1N=V(IJ)	
1105	W1N=W(IJ)	
1106	DO 301 J=2,NJM	
1107	IJ=IMNJ(I)+J	
1108	U1S=U1N	
1109	V1S=V1N	
1110	W1S=W1N	
1111	U1E=U(IJ+NJ)*FX(IJ)+U(IJ)*FX(IJ)	
1112	V1E=V(IJ+NJ)*FY(IJ)+V(IJ)*FY(IJ)	
1113	W1E=W(IJ+NJ)*FZ(IJ)+W(IJ)*FZ(IJ)	
1114	U1N=U(IJ+1)*FY(IJ)+U(IJ)*FY(IJ)	
1115	V1N=V(IJ+1)*FY(IJ)+V(IJ)*FY(IJ)	
1116	W1N=W(IJ+1)*FY(IJ)+W(IJ)*FY(IJ)	
1117 C		
1118	DU1(IJ)=U1E-U1W(IJ)	
1119	DU2(IJ)=U1N-U1S	
1120	DV1(IJ)=V1E-V1W(IJ)	
1121	DV2(IJ)=V1N-V1S	
1122	DW1(IJ)=W1E-W1W(IJ)	
1123	DW2(IJ)=W1N-W1S	
1124 C		
1125	U1W(IJ)=U1E	
1126	V1W(IJ)=V1E	
1127	W1W(IJ)=W1E	
1128 301	CONTINUE	
1129 300	CONTINUE	
1130 C		
1131 C		
1132 C-----	CALCULATE DU2EW(IJ) APPEARING IN THE U-MOMENTUM	
1133 C		
1134	DO 321 I=2,NIM	
1135	IJ=IMNJ(I)+1	
1136	U1N=U2(IJ)*R(IJ)*DEN(IJ)	

Oct 12 1996 16:39

rsmmod_2d

Page 17

```

1137 UYN=UV(IJ)*R(IJ)*DEN(IJ)
1138 VYN=V2(IJ)*R(IJ)*DEN(IJ)
1139 UYN=UM(IJ)*R(IJ)*R(IJ)*DEN(IJ)
1140 VYN=VM(IJ)*R(IJ)*R(IJ)*DEN(IJ)
1141 C
1142 DO 320 J=2,NJM
1143 IJ=IMNJ(I)+J
1144 UUS=UIN
1145 VVS=VIN
1146 UVS=UVN
1147 UWS=UWN
1148 VWS=VWN
1149 IJP=IJ+NJ
1150 IJP=IJ+1
1151 IJN=IJ-1
1152 IJN=IJ-NJ
1153 FXP=FX(IJ)
1154 FYN=FY(IJ)
1155 FYS=1.-FYN
1156 DXE=XX(IJ)-XX(IJN)
1157 DYE=YY(IJ)-YY(IJN)
1158 DXN=XX(IJ)-XX(IJN)
1159 DYN=YY(IJ)-YY(IJN)
1160 RE=HAF*(R(IJ)+R(IJN))
1161 RN=HAF*(R(IJ)+R(IJN))
1162 DYEP=HAF*(YY(IJ)-YY(IJN)+YY(IJN)-YY(IJN-1))
1163 DXEP=HAF*(XX(IJ)-XX(IJN)+XX(IJN)-XX(IJN-1))
1164 DXEP=HAF*(XX(IJ)-XX(IJN)+XX(IJN)-XX(IJN-1))
1165 DXEP=HAF*(XX(IJ)-XX(IJN)+XX(IJN)-XX(IJN-1))
1166 DXEP=HAF*(XX(IJ)-XX(IJN)+XX(IJN)-XX(IJN-1))
1167 DXEP=HAF*(XX(IJ)-XX(IJN)+XX(IJN)-XX(IJN-1))
1168 DXEP=HAF*(XX(IJ)-XX(IJN)+XX(IJN)-XX(IJN-1))
1169 DXN=HAF*(XX(IJ)+XX(IJN)-XX(IJN)-XX(IJN-1))
1170 RP=QTR*(R(IJ)+R(IJN)+R(IJN)-R(IJN-1))
1171 RP=QTR*(R(IJ)+R(IJN)+R(IJN)-R(IJN-1))
1172 RP=QTR*(R(IJ)+R(IJN)+R(IJN)-R(IJN-1))
1173 RP=QTR*(R(IJ)+R(IJN)+R(IJN)-R(IJN-1))
1174 DXE=HAF*(XX(IJ)+XX(IJN)-XX(IJN)-XX(IJN-1))
1175 DYE=HAF*(YY(IJ)+XX(IJN)-XX(IJN)-XX(IJN-1))
1176 DYN=HAF*(YY(IJ)+XX(IJN)-XX(IJN)-XX(IJN-1))
1177 DXN=HAF*(XX(IJ)+XX(IJN)-XX(IJN)-XX(IJN-1))
1178 XP=QTR*(XX(IJ)+XX(IJN)+XX(IJN)-XX(IJN-1))
1179 YE=QTR*(YY(IJ)+XX(IJN)+XX(IJN)-XX(IJN-1))
1180 XE=QTR*(XX(IJ)+XX(IJN)+XX(IJN)-XX(IJN-1))
1181 XN=QTR*(XX(IJ)+XX(IJN)+XX(IJN)-XX(IJN-1))
1182 YN=QTR*(YY(IJ)+XX(IJN)+XX(IJN)-XX(IJN-1))
1183 YN=QTR*(YY(IJ)+XX(IJN)+XX(IJN)-XX(IJN-1))
1184 C
1185 RDENN=R(IJ)*DEN(IJ)*FXW+R(IJN)*DEN(IJN)*FXE
1186 RDENN=R(IJ)*DEN(IJ)*FYS+R(IJN)*DEN(IJN)*FYN
1187 RDENN=R(IJ)*R(IJ)*DEN(IJ)*FXW+R(IJN)*R(IJN)*DEN(IJN)*FXE
1188 RDENN=R(IJ)*R(IJ)*DEN(IJ)*FYS+R(IJN)*R(IJN)*DEN(IJN)*FYN
1189 DEN=DEN(IJ)*FXW+DEN(IJN)*FXE
1190 DEN=DEN(IJ)*FYS+DEN(IJN)*FYN
1191 C
1192 D1P=DYEP*APUU(IJ)*RP
1193 D2P=DYEP*APUU(IJ)*RP
1194 E1P=DXEP*APUU(IJ)*RP
1195 E2P=DXEP*APUU(IJ)*RP
1196 C
1197 D1PE=DYEP*APUU(IJ)*RPE
1198 D2PE=DYEP*APUU(IJ)*RPE
1199 E1PE=DXEP*APUU(IJ)*RPE
1200 E2PE=DXEP*APUU(IJ)*RPE
1201 C
1202 D1PN=DYEN*APUU(IJ)*RPN
1203 D2PN=DYEN*APUU(IJ)*RPN
1204 E1PN=DXEN*APUU(IJ)*RPN
1205 E2PN=DXEN*APUU(IJ)*RPN
1206 C
1207 APUU=APUU(IJ)*FXW+APUU(IJN)*FXE

```

rsmmod_2d

Oct 12 1996 16:39

rsmmod_2d

Page 18

```

1208 D1E=-APUE*DYE*RE
1209 D2E=APUE*(YE-YP)*RE
1210 E1E=-APUE*(XE-XP)*RE
1211 E2E=APUE*DXE*RE
1212 C
1213 APUN=APUU(IJ)*FYS+APUU(IJN)*FYN
1214 D1N=-APUN*(YN-YP)*RN
1215 D2N=APUN*DXN*RN
1216 E1N=-APUN*(XN-XP)*RN
1217 E2N=APUN*(XN-XP)*RN
1218 C
1219 UPE=U(IJN)-U(IJ)
1220 VPE=V(IJN)-V(IJ)
1221 WPE=W(IJN)-W(IJ)
1222 UPN=U(IJN)-U(IJ)
1223 VPN=V(IJN)-V(IJ)
1224 WPN=W(IJN)-W(IJ)
1225 C
1226 C--- CONSIDER TERMS IN UDE-EQ.-----
1227 AA=1.-2./3.*C2+1./3.*C2*C2P*(FUNY(IJ)+4.*FUNX(IJ))
1228 AAB=1.-2./3.*C2+1./3.*C2*C2P*(FUNY(IJ)+4.*FUNX(IJ))
1229 AAN=1.-2./3.*C2+1./3.*C2*C2P*(FUNY(IJ)+4.*FUNX(IJ))
1230 BB=1./3.*C2-2./3.*C2*C2P*(FUNX(IJ)+FUNY(IJ))
1231 BBN=1./3.*C2-2./3.*C2*C2P*(FUNX(IJ)+FUNY(IJ))
1232 CC=1./3.*C2*(1.+C2P*(FUNY(IJ)-2.*FUNX(IJ)))
1233 CCN=1./3.*C2*(1.+C2P*(FUNY(IJ)-2.*FUNX(IJ)))
1234 DD=C2*C2P*(FUNX(IJ))
1235 DDE=C2*C2P*(FUNX(IJ))
1236 DDN=C2*C2P*(FUNX(IJ))
1237 C
1238 TA1P=2.*DEN(IJ)*AA*U2(IJ)*D1P*DU1(IJ)
1239 TA1P=2.*DEN(IJ)*AA*U2(IJ)*D1P*DU1(IJ)
1240 TA2P=2.*DEN(IJ)*AA*U2(IJ)*D1P*DU1(IJ)
1241 TA2P=2.*DEN(IJ)*AA*U2(IJ)*D1P*DU1(IJ)
1242 TA2P=2.*DEN(IJ)*AA*U2(IJ)*D1P*DU1(IJ)
1243 TB1P=2.*DEN(IJ)*BB*U2(IJ)*D1P*DU1(IJ)
1244 TB1P=2.*DEN(IJ)*BB*U2(IJ)*D1P*DU1(IJ)
1245 TB2P=2.*DEN(IJ)*BB*U2(IJ)*D1P*DU1(IJ)
1246 TB2P=2.*DEN(IJ)*BB*U2(IJ)*D1P*DU1(IJ)
1247 TC1P=2.*DEN(IJ)*CC*U2(IJ)*D1P*DU1(IJ)
1248 TC1P=2.*DEN(IJ)*CC*U2(IJ)*D1P*DU1(IJ)
1249 TC2P=2.*DEN(IJ)*CC*U2(IJ)*D1P*DU1(IJ)
1250 TC2P=2.*DEN(IJ)*CC*U2(IJ)*D1P*DU1(IJ)
1251 TD1P=2.*DEN(IJ)*DD*U2(IJ)*D1P*DU1(IJ)
1252 TD1P=2.*DEN(IJ)*DD*U2(IJ)*D1P*DU1(IJ)
1253 TD1P=2.*DEN(IJ)*DD*U2(IJ)*D1P*DU1(IJ)
1254 TD2P=2.*DEN(IJ)*DD*U2(IJ)*D1P*DU1(IJ)
1255 TD2P=2.*DEN(IJ)*DD*U2(IJ)*D1P*DU1(IJ)
1256 C
1257 SA1E=2.*DEN(IJ)*AA*U2(IJ)*D1P*FXW+
1258 SA2E=2.*DEN(IJ)*AA*U2(IJ)*D1P*FXE
1259 SB1E=2.*DEN(IJ)*BB*U2(IJ)*D1P*FXW+
1260 SB2E=2.*DEN(IJ)*BB*U2(IJ)*D1P*FXE
1261 SC1E=2.*DEN(IJ)*CC*U2(IJ)*D1P*FXW+
1262 SC2E=2.*DEN(IJ)*CC*U2(IJ)*D1P*FXE
1263 SD1E=2.*DEN(IJ)*DD*U2(IJ)*D1P*FXW+
1264 SD2E=2.*DEN(IJ)*DD*U2(IJ)*D1P*FXE
1265 SD1E=2.*DEN(IJ)*DD*U2(IJ)*D1P*FXW+
1266 SD2E=2.*DEN(IJ)*DD*U2(IJ)*D1P*FXE
1267 UUE=U2(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
1268 TC2P-TD1E-TD2P)*FXW+
1269 UUE=U2(IJ)-TA1E-TA2E-TB1E-TB2E-TC1E-
1270 TC2E-TD1E-TD2E)*FXE+
1271 SA1E+UPE+SA2E+UPE+SB1E+SB2E+VPE+
1272
1273 C
1274
1275
1276
1277
1278

```

Oct 12 1996 16:39

rsmmod_2d

Page 19

```

1279 C      & SC1E=WPE*SC2E*WPE*SD1E*VPE*SD2E*UPE
1280 C      UUE=UUE*RDENE
1281 C
1282 C-----CONSIDER TERMS IN UUN-EQ.-----
1283 C
1284 C      TA1P=2.*DEN(IJ)*AA*U2(IJ)*D2P*DU2(IJ)
1285      TA1N=2.*DEN(IJ)*AA*U2(IJ)*D2P*DU2(IJ)
1286      TA2P=2.*DEN(IJ)*AA*U2(IJ)*E1P*DU2(IJ)
1287      TA2N=2.*DEN(IJ)*AA*U2(IJ)*E1P*DU2(IJ)
1288      TB1P=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV2(IJ)
1289      TB1N=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV2(IJ)
1290      TB2P=2.*DEN(IJ)*BB*U2(IJ)*E1P*DV2(IJ)
1291      TB2N=2.*DEN(IJ)*BB*U2(IJ)*E1P*DV2(IJ)
1292      TC1P=2.*DEN(IJ)*CC*U2(IJ)*D2P*DW2(IJ)
1293      TC1N=2.*DEN(IJ)*CC*U2(IJ)*D2P*DW2(IJ)
1294      TC2P=2.*DEN(IJ)*CC*U2(IJ)*E1P*DW2(IJ)
1295      TC2N=2.*DEN(IJ)*CC*U2(IJ)*E1P*DW2(IJ)
1296      TD1P=2.*DEN(IJ)*DD*U2(IJ)*D2P*DV2(IJ)
1297      TD1N=2.*DEN(IJ)*DD*U2(IJ)*D2P*DV2(IJ)
1298      TD2P=2.*DEN(IJ)*DD*U2(IJ)*E1P*DV2(IJ)
1299      TD2N=2.*DEN(IJ)*DD*U2(IJ)*E1P*DV2(IJ)
1300      TD2N=2.*DEN(IJ)*DD*U2(IJ)*E1P*DV2(IJ)
1301 C
1302 C      SA1N=2.*DEN(IJ)*AA*U2(IJ)*D2P*FYS +
1303      & 2.*DEN(IJ)*AA*U2(IJ)*D2P*FYN
1304      SA2N=2.*DEN(IJ)*AA*U2(IJ)*E1P*FYS +
1305      & 2.*DEN(IJ)*AA*U2(IJ)*E1P*FYN
1306      SB1N=2.*DEN(IJ)*BB*U2(IJ)*D2P*FYS +
1307      & 2.*DEN(IJ)*BB*U2(IJ)*D2P*FYN
1308      SB2N=2.*DEN(IJ)*BB*U2(IJ)*E1P*FYS +
1309      & 2.*DEN(IJ)*BB*U2(IJ)*E1P*FYN
1310      SC1N=2.*DEN(IJ)*CC*U2(IJ)*D2P*FYS +
1311      & 2.*DEN(IJ)*CC*U2(IJ)*D2P*FYN
1312      SC2N=2.*DEN(IJ)*CC*U2(IJ)*E1P*FYS +
1313      & 2.*DEN(IJ)*CC*U2(IJ)*E1P*FYN
1314      SD1N=2.*DEN(IJ)*DD*U2(IJ)*D2P*FYS +
1315      & 2.*DEN(IJ)*DD*U2(IJ)*D2P*FYN
1316      SD2N=2.*DEN(IJ)*DD*U2(IJ)*E1P*FYS +
1317      & 2.*DEN(IJ)*DD*U2(IJ)*E1P*FYN
1318 C
1319 C      UUN=(U2(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
1320      & TC2P-TD1P-TD2P)*FYS +
1321      & (U2(IJ)-TA1N-TA2N-TB1N-TB2N-TC1N-
1322      & TC2N-TD1N-TD2N)*FYN +
1323      & SA1N*UPN+SA2N*UPN+SB1N*UPN+SB2N*UPN+
1324      & SC1N*UPN+SC2N*UPN+SD1N*UPN+SD2N*UPN
1325 C
1326 C      UUN=UUN*RDENN
1327 C
1328 C-----CONSIDER TERMS IN UVE-EQ.-----
1329 C
1330 C      D1P=-DYPE*APUV(IJ)*RP
1331      D2P=-DYKE*APUV(IJ)*RP
1332      E1P=-DXKE*APUV(IJ)*RP
1333      E2P=-DXPE*APUV(IJ)*RP
1334 C
1335 C      D1PE=-DYPE*APUV(IJ)*RPE
1336      D2PE=-DYKE*APUV(IJ)*RPE
1337      E1PE=-DXKE*APUV(IJ)*RPE
1338      E2PE=-DXPE*APUV(IJ)*RPE
1339 C
1340 C      D1PN=-DYEN*APUV(IJ)*RPN
1341      D2PN=-DYKN*APUV(IJ)*RPN
1342      E1PN=-DXKN*APUV(IJ)*RPN
1343      E2PN=-DXEN*APUV(IJ)*RPN
1344 C
1345 C      APUVE=APUV(IJ)*FXW*APUV(IJ)*FYE
1346      D1E=-APUVE*DYE*RE
1347      D2E=APUVE*(YE-YP)*RE
1348      E1E=-APUVE*(XE-XP)*RE
1349      E2E=APUVE*DYE*RE

```

rsmmod_2d

Oct 12 1996 16:39

rsmmod_2d

Page 20

```

1350 C
1351 C      APUVN=APUV(IJ)*FYS*APUV(IJ)*FYN
1352      DIN=-APUVN*(YN-YP)*RN
1353      DZN=-APUVN*(DYN*RN
1354      EIN=-APUVN*(DXN*RN
1355      EZN=-APUVN*(XN-XP)*RN
1356 C
1357 C      AA=HAF*C2*C2P*FUNXY(IJ)
1358      AAE=HAF*C2*C2P*FUNXY(IJ)
1359      AAN=HAF*C2*C2P*FUNXY(IJ)
1360      BB=AA
1361      BEN=AA
1362      CC=-C2*C2P*FUNXY(IJ)
1363      CCE=-C2*C2P*FUNXY(IJ)
1364      CCN=-C2*C2P*FUNXY(IJ)
1365      DD=1.-C2+1.5*C2*C2P*(FUNX(IJ)+FUNY(IJ))
1366      DDE=1.-C2+1.5*C2*C2P*(FUNX(IJ)+FUNY(IJ))
1367      DDN=1.-C2+1.5*C2*C2P*(FUNX(IJ)+FUNY(IJ))
1368 C
1369 C      TA1P=2.*DEN(IJ)*AA*U2(IJ)*D1P*DU1(IJ)
1370      TA1E=2.*DEN(IJ)*AA*U2(IJ)*D1P*DU1(IJ)
1371      TA2P=2.*DEN(IJ)*AA*U2(IJ)*E2P*DU1(IJ)
1372      TA2E=2.*DEN(IJ)*AA*U2(IJ)*E2P*DU1(IJ)
1373      TB1P=2.*DEN(IJ)*BB*U2(IJ)*D1P*DV1(IJ)
1374      TB1E=2.*DEN(IJ)*BB*U2(IJ)*D1P*DV1(IJ)
1375      TB2P=2.*DEN(IJ)*BB*U2(IJ)*E2P*DV1(IJ)
1376      TB2E=2.*DEN(IJ)*BB*U2(IJ)*E2P*DV1(IJ)
1377      TC1P=2.*DEN(IJ)*CC*U2(IJ)*D1P*DW1(IJ)
1378      TC1E=2.*DEN(IJ)*CC*U2(IJ)*D1P*DW1(IJ)
1379      TC2P=2.*DEN(IJ)*CC*U2(IJ)*E2P*DW1(IJ)
1380      TC2E=2.*DEN(IJ)*CC*U2(IJ)*E2P*DW1(IJ)
1381      TD1P=2.*DEN(IJ)*DD*U2(IJ)*D1P*DV1(IJ)
1382      TD1E=2.*DEN(IJ)*DD*U2(IJ)*D1P*DV1(IJ)
1383      TD2P=2.*DEN(IJ)*DD*U2(IJ)*E2P*DV1(IJ)
1384      TD2E=2.*DEN(IJ)*DD*U2(IJ)*E2P*DV1(IJ)
1385 C
1386 C      SA1E=2.*DEN(IJ)*AA*U2(IJ)*D1P*FXW +
1387      & 2.*DEN(IJ)*AA*U2(IJ)*D1P*FYE
1388      SA2E=2.*DEN(IJ)*AA*U2(IJ)*E2P*FXW +
1389      & 2.*DEN(IJ)*AA*U2(IJ)*E2P*FYE
1390      SB1E=2.*DEN(IJ)*BB*U2(IJ)*D1P*FXW +
1391      & 2.*DEN(IJ)*BB*U2(IJ)*D1P*FYE
1392      SB2E=2.*DEN(IJ)*BB*U2(IJ)*E2P*FXW +
1393      & 2.*DEN(IJ)*BB*U2(IJ)*E2P*FYE
1394      SC1E=2.*DEN(IJ)*CC*U2(IJ)*D1P*FXW +
1395      & 2.*DEN(IJ)*CC*U2(IJ)*D1P*FYE
1396      SC2E=2.*DEN(IJ)*CC*U2(IJ)*E2P*FXW +
1397      & 2.*DEN(IJ)*CC*U2(IJ)*E2P*FYE
1398      SD1E=2.*DEN(IJ)*DD*U2(IJ)*D1P*FXW +
1399      & 2.*DEN(IJ)*DD*U2(IJ)*D1P*FYE
1400      SD2E=2.*DEN(IJ)*DD*U2(IJ)*E2P*FXW +
1401      & 2.*DEN(IJ)*DD*U2(IJ)*E2P*FYE
1402 C
1403 C      UVE=(U2(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
1404      & TC2P-TD1P-TD2P)*FXW +
1405      & (U2(IJ)-TA1E-TA2E-TB1E-TB2E-TC1E-
1406      & TC2E-TD1E-TD2E)*FYE +
1407      & SA1E*UPE+SA2E*UPE+SB1E*UPE+SB2E*UPE+
1408      & SC1E*UPE+SC2E*UPE+SD1E*UPE+SD2E*UPE
1409      & UVE=UVE*RDENE
1410 C
1411 C      UVE=UVE*RDENE
1412 C
1413 C-----CONSIDER TERMS IN UVN-EQ.-----
1414 C
1415 C      TA1P=2.*DEN(IJ)*AA*U2(IJ)*D2P*DU2(IJ)
1416      TA1N=2.*DEN(IJ)*AA*U2(IJ)*D2P*DU2(IJ)
1417      TA2P=2.*DEN(IJ)*AA*U2(IJ)*E1P*DU2(IJ)
1418      TA2N=2.*DEN(IJ)*AA*U2(IJ)*E1P*DU2(IJ)
1419      TB1P=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV2(IJ)
1420      TB1N=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV2(IJ)

```

10

```

1492 BBE=1.-2./3.*C2+1./3.*C2+C2P*
1493 & (4.*FUNY(IPJ)+FUNX(IPJ))
1494 BBN=1.-2./3.*C2+1./3.*C2+C2P*
1495 & (4.*FUNY(IJP)+FUNX(IJP))
1496 CC=1./3.*C2+1./3.*C2+C2P*(FUNX(IJ)-2.*FUNY(IJ))
1497 CCN=1./3.*C2+1./3.*C2+C2P*(FUNX(IJ)-2.*FUNY(IJ))
1498 DD=C2+C2P*(FUNX(IJ)-2.*FUNY(IJ))
1499 DDE=C2+C2P*(FUNX(IJ))
1500 DDN=C2+C2P*(FUNX(IJ))
1501 C
1502 TA1P=2.*DEN(IJ)*AA*U2(IJ)*D1P*DV1(IJ)
1503 TA1E=2.*DEN(IJ)*AA*U2(IJ)*D1P*DV1(IJ)
1504 TA2P=2.*DEN(IJ)*AA*U2(IJ)*D2P*DV1(IJ)
1505 TA2E=2.*DEN(IJ)*AA*U2(IJ)*D2P*DV1(IJ)
1506 TB1P=2.*DEN(IJ)*BB*U2(IJ)*D1P*DV1(IJ)
1507 TB1E=2.*DEN(IJ)*BB*U2(IJ)*D1P*DV1(IJ)
1508 TB2P=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV1(IJ)
1509 TB2E=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV1(IJ)
1510 TC1P=2.*DEN(IJ)*CC*U2(IJ)*D1P*DV1(IJ)
1511 TC1E=2.*DEN(IJ)*CC*U2(IJ)*D1P*DV1(IJ)
1512 TC2P=2.*DEN(IJ)*CC*U2(IJ)*D2P*DV1(IJ)
1513 TC2E=2.*DEN(IJ)*CC*U2(IJ)*D2P*DV1(IJ)
1514 TD1P=2.*DEN(IJ)*DD*U2(IJ)*D1P*DV1(IJ)
1515 TD1E=2.*DEN(IJ)*DD*U2(IJ)*D1P*DV1(IJ)
1516 TD2P=2.*DEN(IJ)*DD*U2(IJ)*D2P*DV1(IJ)
1517 TD2E=2.*DEN(IJ)*DD*U2(IJ)*D2P*DV1(IJ)
1518 C
1519 SA1E=2.*DEN(IJ)*AA*U2(IJ)*D1P*FXW +
1520 & 2.*DEN(IJ)*AA*U2(IJ)*D1P*FXE +
1521 SA2E=2.*DEN(IJ)*AA*U2(IJ)*D2P*FXW +
1522 & 2.*DEN(IJ)*AA*U2(IJ)*D2P*FXE +
1523 SB1E=2.*DEN(IJ)*BB*U2(IJ)*D1P*FXW +
1524 & 2.*DEN(IJ)*BB*U2(IJ)*D1P*FXE +
1525 SB2E=2.*DEN(IJ)*BB*U2(IJ)*D2P*FXW +
1526 & 2.*DEN(IJ)*BB*U2(IJ)*D2P*FXE +
1527 SC1E=2.*DEN(IJ)*CC*U2(IJ)*D1P*FXW +
1528 & 2.*DEN(IJ)*CC*U2(IJ)*D1P*FXE +
1529 SC2E=2.*DEN(IJ)*CC*U2(IJ)*D2P*FXW +
1530 & 2.*DEN(IJ)*CC*U2(IJ)*D2P*FXE +
1531 SD1E=2.*DEN(IJ)*DD*U2(IJ)*D1P*FXW +
1532 & 2.*DEN(IJ)*DD*U2(IJ)*D1P*FXE +
1533 SD2E=2.*DEN(IJ)*DD*U2(IJ)*D2P*FXW +
1534 & 2.*DEN(IJ)*DD*U2(IJ)*D2P*FXE +
1535 VVE=(V2(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
1536 & TC2P-TD1P-TD2P)*FXW +
1537 & (V2(IJ)-TA1E-TA2E-TB1E-TB2E-TC1E-
1538 & TC2E-TD1E-TD2E)*FXE +
1539 & SA1E+UPE+SA2E+UPE+SB1E+VPE+SB2E+VPE+
1540 & SC1E+WPE+SC2E+WPE+SD1E+VPE+SD2E+UPE
1541 C
1542 VVE=VVE*ROENE
1543 C
1544 C
1545 C
1546 C
1547 C---CONSIDER TERMS IN VVN-EQ. ---
1548 C
1549 TA1P=2.*DEN(IJ)*AA*U2(IJ)*D2P*DV2(IJ)
1550 TA1E=2.*DEN(IJ)*AA*U2(IJ)*D2P*DV2(IJ)
1551 TA2P=2.*DEN(IJ)*AA*U2(IJ)*D2P*DV2(IJ)
1552 TA2E=2.*DEN(IJ)*AA*U2(IJ)*D2P*DV2(IJ)
1553 TB1P=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV2(IJ)
1554 TB1E=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV2(IJ)
1555 TB2P=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV2(IJ)
1556 TB2E=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV2(IJ)
1557 TC1P=2.*DEN(IJ)*CC*U2(IJ)*D2P*DV2(IJ)
1558 TC1E=2.*DEN(IJ)*CC*U2(IJ)*D2P*DV2(IJ)
1559 TC2P=2.*DEN(IJ)*CC*U2(IJ)*D2P*DV2(IJ)
1560 TC2E=2.*DEN(IJ)*CC*U2(IJ)*D2P*DV2(IJ)
1561 TD1P=2.*DEN(IJ)*DD*U2(IJ)*D2P*DV2(IJ)
1562 TD1E=2.*DEN(IJ)*DD*U2(IJ)*D2P*DV2(IJ)

```

```

1421 TB2P=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV2(IJ)
1422 TB2E=2.*DEN(IJ)*BB*U2(IJ)*D2P*DV2(IJ)
1423 TC1P=2.*DEN(IJ)*CC*U2(IJ)*D2P*DV2(IJ)
1424 TC1E=2.*DEN(IJ)*CC*U2(IJ)*D2P*DV2(IJ)
1425 TC2P=2.*DEN(IJ)*CC*U2(IJ)*D2P*DV2(IJ)
1426 TC2E=2.*DEN(IJ)*CC*U2(IJ)*D2P*DV2(IJ)
1427 TD1P=2.*DEN(IJ)*DD*U2(IJ)*D2P*DV2(IJ)
1428 TD1E=2.*DEN(IJ)*DD*U2(IJ)*D2P*DV2(IJ)
1429 TD2P=2.*DEN(IJ)*DD*U2(IJ)*D2P*DV2(IJ)
1430 TD2E=2.*DEN(IJ)*DD*U2(IJ)*D2P*DV2(IJ)
1431 C
1432 SAIN=2.*DEN(IJ)*AA*U2(IJ)*D2P*FYS +
1433 & 2.*DEN(IJ)*AA*U2(IJ)*D2P*FYN +
1434 SA2N=2.*DEN(IJ)*AA*U2(IJ)*D2P*FYS +
1435 & 2.*DEN(IJ)*AA*U2(IJ)*D2P*FYN +
1436 SB1N=2.*DEN(IJ)*BB*U2(IJ)*D2P*FYS +
1437 & 2.*DEN(IJ)*BB*U2(IJ)*D2P*FYN +
1438 SB2N=2.*DEN(IJ)*BB*U2(IJ)*D2P*FYS +
1439 & 2.*DEN(IJ)*BB*U2(IJ)*D2P*FYN +
1440 SC1N=2.*DEN(IJ)*CC*U2(IJ)*D2P*FYS +
1441 & 2.*DEN(IJ)*CC*U2(IJ)*D2P*FYN +
1442 SC2N=2.*DEN(IJ)*CC*U2(IJ)*D2P*FYS +
1443 & 2.*DEN(IJ)*CC*U2(IJ)*D2P*FYN +
1444 SD1N=2.*DEN(IJ)*DD*U2(IJ)*D2P*FYS +
1445 & 2.*DEN(IJ)*DD*U2(IJ)*D2P*FYN +
1446 SD2N=2.*DEN(IJ)*DD*U2(IJ)*D2P*FYS +
1447 & 2.*DEN(IJ)*DD*U2(IJ)*D2P*FYN +
1448 C
1449 UVN=(U2(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
1450 & TC2P-TD1P-TD2P)*FYS +
1451 & (U2(IJ)-TA1E-TA2E-TB1E-TB2E-TC1E-
1452 & TC2E-TD1E-TD2E)*FYN +
1453 & SA1N+UPE+SA2N+UPE+SB1N+VPE+SB2N+VPE+
1454 & SC1N+WPE+SC2N+WPE+SD1N+VPE+SD2N+UPE
1455 C
1456 UVN=UVN*ROENE
1457 C
1458 C---CONSIDER TERMS IN VVE-EQ. ---
1459 C
1460 C
1461 D1P=-DYE*APV(IJ)*RP
1462 D2P=DYK*APV(IJ)*RP
1463 E1P=-DXK*APV(IJ)*RP
1464 E2P=DXE*APV(IJ)*RP
1465 C
1466 D1PE=-DYE*APV(IJ)*RPE
1467 D2PE=DYK*APV(IJ)*RPE
1468 E1PE=-DXK*APV(IJ)*RPE
1469 E2PE=DXE*APV(IJ)*RPE
1470 C
1471 D1PN=-DYE*APV(IJ)*RPN
1472 D2PN=DYK*APV(IJ)*RPN
1473 E1PN=-DXK*APV(IJ)*RPN
1474 E2PN=DXE*APV(IJ)*RPN
1475 C
1476 APVVE=APV(IJ)*FXW*APV(IJ)*FXE
1477 D1E=APVVE*DY*RE
1478 D2E=APVVE*(YE-YF)*RE
1479 E1E=APVVE*(XE-XP)*RE
1480 E2E=APVVE*(XE-XP)*RE
1481 C
1482 APVNV=APV(IJ)*FYS*APV(IJ)*FYN
1483 D1N=APVNV*(YN-YF)*RN
1484 D2N=APVNV*(YN-YF)*RN
1485 E1N=APVNV*(XE-XP)*RN
1486 E2N=APVNV*(XE-XP)*RN
1487 C
1488 AA=1./3.*C2-2./3.*C2+C2P*(FUNX(IJ)+FUNY(IJ))
1489 AA=1./3.*C2-2./3.*C2+C2P*(FUNX(IJ)+FUNY(IJ))
1490 BB=1.-2./3.*C2+1./3.*C2+C2P*(FUNX(IJ)+FUNY(IJ))
1491 BB=1.-2./3.*C2+1./3.*C2+C2P*(FUNX(IJ)+FUNY(IJ))

```


Oct 12 1996 16:39

rsmmod_2d

Page 23

```

1563 TD2P=DEN(IJ)*DD*V2(IJ)*E1P*DU2(IJ)
1564 TD2N=DEN(IJ)*DDN*V2(IJ)*E1PN*DU2(IJ)
1565 C
1566 SAIN=2.*DEN(IJ)*AA*U2(IJ)*D2P*FYS +
1567 & 2.*DEN(IJ)*AA*U2(IJ)*D2PN*FYN
1568 SAZN=2.*DEN(IJ)*AA*UV(IJ)*E1P*FYS +
1569 & 2.*DEN(IJ)*AA*UV(IJ)*E1PN*FYN
1570 SBIN=2.*DEN(IJ)*BB*UV(IJ)*D2P*FYS +
1571 & 2.*DEN(IJ)*BB*UV(IJ)*D2PN*FYN
1572 SBZN=2.*DEN(IJ)*BB*V2(IJ)*E1P*FYS +
1573 & 2.*DEN(IJ)*BB*V2(IJ)*E1PN*FYN
1574 SCIN=2.*DEN(IJ)*CC*UV(IJ)*D2P*FYS +
1575 & 2.*DEN(IJ)*CC*UV(IJ)*D2PN*FYN
1576 SCZN=2.*DEN(IJ)*CC*VW(IJ)*E1P*FYS +
1577 & 2.*DEN(IJ)*CC*VW(IJ)*E1PN*FYN
1578 SDIN=DEN(IJ)*DD*U2(IJ)*D2P*FYS +
1579 & DEN(IJ)*DD*U2(IJ)*D2PN*FYN
1580 SDZN=DEN(IJ)*DD*V2(IJ)*E1P*FYS +
1581 & DEN(IJ)*DDN*V2(IJ)*E1PN*FYN
1582 C
1583 VVN=(V2(IJ)-TA1P-TA2P-TB1P-TB2P-TC1P-
1584 & TC2P-TD1P-TD2P)*FYS +
1585 & (V2(IJ)-TAIN-TA2N-TBIN-TB2N-TC1N-
1586 & TC2N-TD1N-TD2N)*FYN +
1587 & SAIN*UPN+SAZN*UPN+SBIN*VFN+SBZN*VFN+
1588 & SCIN*WPN+SC2N*WPN+SDIN*VFN+SD2N*UPN
1589 C
1590 VVN=VVN*RDENN
1591 C
1592 C
1593 C---CONSIDER TERMS IN UWE-EQ. ---
1594 C
1595 D1P=-DYE*APUW(IJ)*RP
1596 D2P=DKP*APUW(IJ)*RP
1597 E1P=-DXK*APUW(IJ)*RP
1598 E2P=DXEP*APUW(IJ)*RP
1599 C
1600 D1PE=-DYE*APUW(IJ)*RPE
1601 D2PE=DKP*APUW(IJ)*RPE
1602 E1PE=-DXKE*APUW(IJ)*RPE
1603 E2PE=DXEP*APUW(IJ)*RPE
1604 C
1605 D1PN=-DYN*APUW(IJ)*RPN
1606 D2PN=DKN*APUW(IJ)*RPN
1607 E1PN=-DXKN*APUW(IJ)*RPN
1608 E2PN=DXEN*APUW(IJ)*RPN
1609 C
1610 APUW=APUW(IJ)*FXW*APUW(IJ)*FYE
1611 D1E=-APUW*DY*RE
1612 D2E=APUW*(YE-YF)*RE
1613 E1E=-APUW*(XE-XP)*RE
1614 E2E=APUW*(XE-XP)*RE
1615 C
1616 APUW=APUW(IJ)*FYS*APUW(IJ)*FYN
1617 D1N=-APUW*(YN-YP)*RN
1618 D2N=APUW*(YN-YP)*RN
1619 E1N=-APUW*(XN-XP)*RN
1620 E2N=APUW*(XN-XP)*RN
1621 C
1622 AA=1.-C2+1.5*C2*C2P*FUNX(IJ)
1623 AA=1.-C2+1.5*C2*C2P*FUNX(IJ)
1624 AA=1.-C2+1.5*C2*C2P*FUNX(IJ)
1625 BB=1.5*C2*C2P*FUNX(IJ)
1626 BB=1.5*C2*C2P*FUNX(IJ)
1627 BB=1.5*C2*C2P*FUNX(IJ)
1628 C
1629 TAIP=AA*DEN(IJ)*U2(IJ)*D1P*DW1(IJ)
1630 TAIE=AAE*DEN(IJ)*U2(IJ)*D1PE*DW1(IJ)
1631 TA2P=AA*DEN(IJ)*UV(IJ)*E2P*DW1(IJ)
1632 TA2E=AAE*DEN(IJ)*UV(IJ)*E2PE*DW1(IJ)
1633 TA3P=AA*DEN(IJ)*VW(IJ)*E2P*DW1(IJ)

```

rsmmod_2d

12

Oct 12 1996 16:39

rsmmod_2d

Page 24

```

1634 TA3E=AAE*DEN(IJ)*VW(IJ)*E2PE*DW1(IJ)
1635 TA4P=AA*DEN(IJ)*UW(IJ)*E2P*DW1(IJ)
1636 TA4E=AAE*DEN(IJ)*UW(IJ)*E2PE*DW1(IJ)
1637 TB1P=BB*DEN(IJ)*UV(IJ)*D1P*DW1(IJ)
1638 TB1E=BBE*DEN(IJ)*UV(IJ)*D1PE*DW1(IJ)
1639 TB2P=BB*DEN(IJ)*UW(IJ)*D1P*DW1(IJ)
1640 TB2E=BBE*DEN(IJ)*UW(IJ)*D1PE*DW1(IJ)
1641 TB3P=BB*DEN(IJ)*V2(IJ)*E2P*DW1(IJ)
1642 TB3E=BBE*DEN(IJ)*V2(IJ)*E2PE*DW1(IJ)
1643 TB4P=BB*DEN(IJ)*VW(IJ)*E2P*DW1(IJ)
1644 TB4E=BBE*DEN(IJ)*VW(IJ)*E2PE*DW1(IJ)
1645 C
1646 SA1E=AA*DEN(IJ)*U2(IJ)*D1P*FXW +
1647 & AAE*DEN(IJ)*U2(IJ)*D1PE*FXE
1648 SA2E=AA*DEN(IJ)*UV(IJ)*E2P*FXW +
1649 & AAE*DEN(IJ)*UV(IJ)*E2PE*FXE
1650 SA3E=AA*DEN(IJ)*VW(IJ)*E2P*FXW +
1651 & AAE*DEN(IJ)*VW(IJ)*E2PE*FXE
1652 SA4E=AA*DEN(IJ)*UW(IJ)*E2P*FXW +
1653 & AAE*DEN(IJ)*UW(IJ)*E2PE*FXE
1654 SB1E=BB*DEN(IJ)*UV(IJ)*D1P*FXW +
1655 & BBE*DEN(IJ)*UV(IJ)*D1PE*FXE
1656 SB2E=BB*DEN(IJ)*UW(IJ)*D1P*FXW +
1657 & BBE*DEN(IJ)*UW(IJ)*D1PE*FXE
1658 SB3E=BB*DEN(IJ)*V2(IJ)*E2P*FXW +
1659 & BBE*DEN(IJ)*V2(IJ)*E2PE*FXE
1660 SB4E=BB*DEN(IJ)*VW(IJ)*E2P*FXW +
1661 & BBE*DEN(IJ)*VW(IJ)*E2PE*FXE
1662 C
1663 UWE=(UW(IJ)-TA1P-TA2P-TA3P-TA4P-
1664 & TB1P-TB2P-TB3P-TB4P)*FXW +
1665 & (UW(IJ)-TA1E-TA2E-TA3E-TA4E-
1666 & TB1E-TB2E-TB3E-TB4E)*FXE +
1667 & SA1E*WPE+SA2E*WPE+SA3E*UPE+SA4E*VPE+
1668 & SB1E*WPE+SB2E*VPE+SB3E*WPE+SB4E*VPE
1669 C
1670 UWE=UWE*RRDENE
1671 C
1672 C
1673 C---CONSIDER TERMS IN UWN-EQ. ---
1674 C
1675 TA1P=AA*DEN(IJ)*U2(IJ)*D2P*DW2(IJ)
1676 TA1N=AA*DEN(IJ)*U2(IJ)*D2PN*DW2(IJ)
1677 TA2P=AA*DEN(IJ)*UV(IJ)*E1P*DW2(IJ)
1678 TA2N=AA*DEN(IJ)*UV(IJ)*E1PN*DW2(IJ)
1679 TA3P=AA*DEN(IJ)*VW(IJ)*E1P*DW2(IJ)
1680 TA4P=AA*DEN(IJ)*UW(IJ)*E1P*DW2(IJ)
1681 TB1P=BB*DEN(IJ)*UV(IJ)*D2P*DW2(IJ)
1682 TB1N=BB*DEN(IJ)*UV(IJ)*D2PN*DW2(IJ)
1683 TB2P=BB*DEN(IJ)*UW(IJ)*D2P*DW2(IJ)
1684 TB2N=BB*DEN(IJ)*UW(IJ)*D2PN*DW2(IJ)
1685 TB3P=BB*DEN(IJ)*V2(IJ)*E1P*DW2(IJ)
1686 TB3N=BB*DEN(IJ)*V2(IJ)*E1PN*DW2(IJ)
1687 TB4P=BB*DEN(IJ)*VW(IJ)*E1P*DW2(IJ)
1688 TB4N=BB*DEN(IJ)*VW(IJ)*E1PN*DW2(IJ)
1689 C
1690 SA1N=AA*DEN(IJ)*U2(IJ)*D2P*FYS +
1691 & AAN*DEN(IJ)*U2(IJ)*D2PN*FYN
1692 SA2N=AA*DEN(IJ)*UV(IJ)*E1P*FYS +
1693 & AAN*DEN(IJ)*UV(IJ)*E1PN*FYN
1694 SA3N=AA*DEN(IJ)*VW(IJ)*E1P*FYS +
1695 & AAN*DEN(IJ)*VW(IJ)*E1PN*FYN
1696 SA4N=AA*DEN(IJ)*UW(IJ)*E1P*FYS +
1697 & AAN*DEN(IJ)*UW(IJ)*E1PN*FYN
1698 SB1N=BB*DEN(IJ)*UV(IJ)*D2P*FYS +
1699 & BBN*DEN(IJ)*UV(IJ)*D2PN*FYN
1700 SB2N=BB*DEN(IJ)*UW(IJ)*D2P*FYS +
1701 & BBN*DEN(IJ)*UW(IJ)*D2PN*FYN
1702 SB3N=BB*DEN(IJ)*V2(IJ)*E1P*FYS +
1703 & BBN*DEN(IJ)*V2(IJ)*E1PN*FYN
1704 SB4N=BB*DEN(IJ)*VW(IJ)*E1P*FYS +

```


Oct 12 1996 16:39	rsmod_2d	Page 27
1847	RRW=HAF*(R(IMJ)+R(IMJ-1))	
1848	RS=HAF*(R(IJM)+R(IMJ-1))	
1849 C		
1850	DU2EW(IJ)=UUE-UUW(J)	
1851	DU2NS(IJ)=UUN-UUS	
1852	DVUNS(IJ)=UVN-UVS	
1853	DVVEW(IJ)=UVE-VVW(J)	
1854	DV2EW(IJ)=VVE-VVW(J)	
1855	DV2NS(IJ)=VVN-VVS	
1856	DUMEW(IJ)=UWE-UWM(J)	
1857	DUMNS(IJ)=UWN-UWS	
1858	DVWEW(IJ)=VWE-VVW(J)	
1859	DVWNS(IJ)=VWN-VVS	
1860 C		
1861	UUW(J)=UUE	
1862	UVW(J)=UVE	
1863	VVM(J)=VVE	
1864	UWM(J)=UWE	
1865	VWM(J)=VWE	
1866 C		
1867 320	CONTINUE	
1868 321	CONTINUE	
1869 C		
1870	DO 330 I=2,NJM	
1871	DO 330 J=2,NJM	
1872	IJ=IMNJ(I)+J	
1873	IMJ=IJ-NJ	
1874	IJM=IJ-1	
1875	RP=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IMJ-1))	
1876	RP=QTR*(R(IJ)+R(IMJ)+R(IJM)+R(IMJ-1))	
1877	DP2EW(IJ)=DU2EW(IJ)/RP	
1878	DP2NS(IJ)=DU2NS(IJ)/RP	
1879	DPV2EW(IJ)=DV2EW(IJ)/RP	
1880 330	CONTINUE	
1881 C		
1882	RETURN	
1883	END	
1884 C		
1885 C		
1886 C		
1887	SUBROUTINE MODIJJ (ITBS,ITEN,JTBW,JTBE)	
1888	CONTINUE	
1889	INCLUDE 'gridparam.h'	
1890	INCLUDE 'rsm.h'	
1891	DIMENSION ITBS(NX),ITEN(NX),JTBW(NY),JTBE(NY)	
1892 C		
1893	NI=NIM+1	
1894	NJ=NMJ+1	
1895 C		
1896 C	SOUTHE WALL	
1897 C		
1898	DO 1110 I=2,NIM	
1899	IJ=IMNJ(I)+2	
1900	IF (ITBS(I).EQ.4) THEN	
1901	GEN(IJ)=GENTS(I)	
1902	ENDIF	
1903 C		
1904 C	NORTH WALL	
1905 C		
1906	IJ=IMNJ(I)+NMJ	
1907	IF (ITEN(I).EQ.4) THEN	
1908	GEN(IJ)=GENTN(I)	
1909	ENDIF	
1910 C		
1911 1110	CONTINUE	
1912 C		
1913 C	WEST WALL	
1914 C		
1915	DO 1120 J=2,NJM	
1916	IJ=IMNJ(2)+J	
1917	IF (JTBW(J).EQ.4) THEN	

Oct 12 1996 16:39	rsmod_2d	Page 28
1918	GEN(IJ)=GENTW(J)	
1919	ENDIF	
1920 C		
1921 C	EAST WALL	
1922 C		
1923	IJ=IMNJ(NIM)+J	
1924	IF (JTBE(J).EQ.4) THEN	
1925	GEN(IJ)=GENTEE(J)	
1926	ENDIF	
1927 C		
1928 1120	CONTINUE	
1929 C		
1930	RETURN	
1931	END	
1932 C		
1933 C		
1934	SUBROUTINE MODUIUJ (PHI,IPHI,X,Y,FX,FY,ARE,VOL,R,	
1935	& DEN,TE,ED,ITBS,ITEN,JTBE,JTBW,	
1936	& DNS,DNN,DNE,DNW)	
1937 C		
1938	INCLUDE 'gridparam.h'	
1939	INCLUDE 'rsm.h'	
1940	DIMENSION PHI(NXNY),DEN(NXNY),TE(NXNY),ED(NXNY)	
1941	DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY),ARE(NXNY)	
1942	# VOL(NXNY),R(NXNY)	
1943	DIMENSION ITBS(NX),ITEN(NX),JTBW(NY),JTBE(NY)	
1944	DIMENSION DNS(NX),DNN(NX),DNE(NY),DNW(NY)	
1945 C		
1946 C	CONSTANTS FOR THE REYNOLDS STRESSES NEAR THE WALL	
1947 C	TAKEN FROM (LEIN @ LESCHZNER)	
1948 C	CUU=1.098	
1949 C	CVV=0.247	
1950 C	CWW=0.655	
1951 C	CUV=-0.255	
1952 C	CVM= 0.0	
1953 C	CWN= 0.0	
1954 C	VALUES OF CVM AND CWN ARE SET TO 0.0	
1955 C	FOR LACK OF BETTER VALUES !!	
1956 C		
1957 C	UPDATE -TE,-ED,-U2,-V2,-W2,-UV,-VW,-UW- B.C'S-	
1958	GO TO (1200,1300,1400,1500,1600,1700,1800,1900) IDIR	
1959 C		
1960 C	TURBULENT KINETIC ENERGY BOUNDARY CONDITIONS	
1961 C		
1962 1200	CONTINUE	
1963 C	SOUTH BOUNDARY	
1964	DO 1210 I=2,NIM	
1965	IJ=IMNJ(I)+2	
1966	GO TO (1211,1212,1213) ITBS(I)	
1967 1211	CONTINUE	
1968	SU(IJ)=SU(IJ)+AS(IJ)*TE(IJ-1)	
1969	BP(IJ)=BP(IJ)+AS(IJ)	
1970	GO TO 1212	
1971 1213	CONTINUE	
1972	GEN(IJ)=GENTS(I)	
1973	RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))	
1974	SU(IJ)=APV(IJ)*GEN(IJ)*VOL(IJ)	
1975 1212	CONTINUE	
1976	AS(IJ)=0.0	
1977 1210	CONTINUE	
1978 C	NORTH BOUNDARY	
1979	DO 1220 I=2,NIM	
1980	IJ=IMNJ(I)+NMJ	
1981	GO TO (1221,1222,1223) ITEN(I)	
1982 1221	CONTINUE	
1983	SU(IJ)=SU(IJ)+AN(IJ)*TE(IJ+1)	
1984	BP(IJ)=BP(IJ)+AN(IJ)	
1985	GO TO 1222	
1986 1223	CONTINUE	
1987	GEN(IJ)=GENTN(I)	
1988	RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))	

Oct 12 1996 16:39	rsmod_2d	Page 29
1989	SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)	
1990	CONTINUE	
1991	AN(IJ)=0.0	
1992	CONTINUE	
1993	C-----WEST BOUNDARY	
1994	DO 1230 J=2,NJM	
1995	IJ=IMNJ(2)+J	
1996	GO TO (1231,1232,1233,1233) JTBW(J)	
1997	CONTINUE	
1998	SU(IJ)=SU(IJ)+AW(IJ)*TE(IJ-NJ)	
1999	BP(IJ)=BP(IJ)+AW(IJ)	
2000	GO TO 1232	
2001	CONTINUE	
2002	GEN(IJ)=GENTW(J)	
2003	RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))	
2004	SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)	
2005	CONTINUE	
2006	AW(IJ)=0.0	
2007	CONTINUE	
2008	C-----EAST BOUNDARY	
2009	DO 1240 J=2,NJM	
2010	IJ=IMNJ(NIM)+J	
2011	GO TO (1241,1242,1242,1243) JTBW(J)	
2012	CONTINUE	
2013	SU(IJ)=SU(IJ)+AE(IJ)*TE(IJ-NJ)	
2014	BP(IJ)=BP(IJ)+AE(IJ)	
2015	GO TO 1242	
2016	CONTINUE	
2017	GEN(IJ)=GENTEE(J)	
2018	RP=QTR*(R(IJ)+R(IJ-NJ)+R(IJ-1)+R(IJ-NJ-1))	
2019	SU(IJ)=APV(IJ)+GEN(IJ)*VOL(IJ)	
2020	CONTINUE	
2021	AE(IJ)=0.0	
2022	CONTINUE	
2023	C	
2024	RETURN	
2025	C	
2026	C-----BOUNDARY CONDITIONS FOR TURBULENT ENERGY DISSIPATION	
2027	C	
2028	1300 CONTINUE	
2029	C-----SOUTH BOUNDARY	
2030	DO 1310 I=2,NIM	
2031	IJ=IMNJ(I)+2	
2032	GO TO (1311,1312,1312,1313) ITBS(I)	
2033	CONTINUE	
2034	SU(IJ)=SU(IJ)+AS(IJ)*ED(IJ-1)	
2035	BP(IJ)=BP(IJ)+AS(IJ)	
2036	GO TO 1312	
2037	CONTINUE	
2038	TE(IJ)=ABS(TE(IJ))	
2039	SU(IJ)=GREAT*CMU75*TE(IJ)*SORT(TE(IJ))/(CAPPA*DNS(I))	
2040	BP(IJ)=GREAT	
2041	CONTINUE	
2042	AS(IJ)=0.0	
2043	CONTINUE	
2044	C-----NORTH BOUNDARY	
2045	DO 1320 I=2,NIM	
2046	IJ=IMNJ(I)+NJM	
2047	GO TO (1321,1322,1322,1323) ITBN(I)	
2048	CONTINUE	
2049	SU(IJ)=SU(IJ)+AN(IJ)*ED(IJ+1)	
2050	BP(IJ)=BP(IJ)+AN(IJ)	
2051	GO TO 1322	
2052	CONTINUE	
2053	TE(IJ)=ABS(TE(IJ))	
2054	SU(IJ)=GREAT*CMU75*TE(IJ)*SORT(TE(IJ))/(CAPPA*DNN(I))	
2055	BP(IJ)=GREAT	
2056	CONTINUE	
2057	AN(IJ)=0.0	
2058	CONTINUE	
2059	C-----WEST BOUNDARY	

Oct 12 1996 16:39	rsmod_2d	Page 30
2060	DO 1330 J=2,NJM	
2061	IJ=IMNJ(2)+J	
2062	GO TO (1331,1332,1332,1333) JTBW(J)	
2063	CONTINUE	
2064	SU(IJ)=SU(IJ)+AW(IJ)*ED(IJ-NJ)	
2065	BP(IJ)=BP(IJ)+AW(IJ)	
2066	GO TO 1332	
2067	CONTINUE	
2068	TE(IJ)=ABS(TE(IJ))	
2069	SU(IJ)=GREAT*CMU75*TE(IJ)*SORT(TE(IJ))/(CAPPA*DNN(J))	
2070	BP(IJ)=GREAT	
2071	CONTINUE	
2072	AW(IJ)=0.0	
2073	CONTINUE	
2074	C-----EAST BOUNDARY	
2075	DO 1340 J=2,NJM	
2076	IJ=IMNJ(NIM)+J	
2077	GO TO (1341,1342,1342,1343) JTBW(J)	
2078	CONTINUE	
2079	SU(IJ)=SU(IJ)+AE(IJ)*ED(IJ-NJ)	
2080	BP(IJ)=BP(IJ)+AE(IJ)	
2081	GO TO 1342	
2082	CONTINUE	
2083	TE(IJ)=ABS(TE(IJ))	
2084	SU(IJ)=GREAT*CMU75*TE(IJ)*SORT(TE(IJ))/(CAPPA*DNE(J))	
2085	BP(IJ)=GREAT	
2086	CONTINUE	
2087	AE(IJ)=0.0	
2088	CONTINUE	
2089	C	
2090	RETURN	
2091	C	
2092	C-----BOUNDARY CONDITIONS FOR U2-REYNOLDS STRESS	
2093	C	
2094	1400 CONTINUE	
2095	C-----SOUTH BOUNDARY	
2096	DO 1410 I=2,NIM	
2097	IJ=IMNJ(I)+2	
2098	GO TO (1411,1412,1412,1413) ITBS(I)	
2099	CONTINUE	
2100	SU(IJ)=SU(IJ)+AS(IJ)*U2(IJ-1)	
2101	BP(IJ)=BP(IJ)+AS(IJ)	
2102	GO TO 1412	
2103	CONTINUE	
2104	TE(IJ)=ABS(TE(IJ))	
2105	U2WAL=CUU*TE(IJ)	
2106	V2WAL=CUV*TE(IJ)	
2107	U2REAL=U2WAL*FTIS(I)**2+V2WAL*FNIS(I)**2+	
2108	& 2.*U2WAL*FTIS(I)*FNIS(I)	
2109	SU(IJ)=GREAT*U2REAL	
2110	BP(IJ)=GREAT	
2111	CONTINUE	
2112	AS(IJ)=0.0	
2113	CONTINUE	
2114	1410 CONTINUE	
2115	C-----NORTH BOUNDARY	
2116	DO 1420 I=2,NIM	
2117	IJ=IMNJ(I)+NJM	
2118	GO TO (1421,1422,1422,1423) ITBN(I)	
2119	CONTINUE	
2120	SU(IJ)=SU(IJ)+AN(IJ)*U2(IJ+1)	
2121	BP(IJ)=BP(IJ)+AN(IJ)	
2122	GO TO 1422	
2123	CONTINUE	
2124	TE(IJ)=ABS(TE(IJ))	
2125	U2WAL=CUU*TE(IJ)	
2126	V2WAL=CUV*TE(IJ)	
2127	U2REAL=U2WAL*FTIN(I)**2+V2WAL*FNIN(I)**2+	
2128	& 2.*U2WAL*FTIN(I)*FNIN(I)	
2129	SU(IJ)=GREAT*U2REAL	
2130		

Oct 12 1996 16:39	rsmod_2d	Page 31
2131	BP(IJ)=GREAT	
2132	CONTINUE	
2133	AN(IJ)=0.	
2134	CONTINUE	
2135	C-----WEST BOUNDARY	
2136	DO 1430 J=2,NJM	
2137	IJ=IMNJ(2)+J	
2138	GO TO (1431,1432,1432,1433) JTBW(J)	
2139	CONTINUE	
2140	SU(IJ)=SU(IJ)+AW(IJ)*U2(IJ-NJ)	
2141	BP(IJ)=BP(IJ)+AW(IJ)	
2142	GO TO 1432	
2143	CONTINUE	
2144	TE(IJ)=ABS(TE(IJ))	
2145	UWAL=CUV*TE(IJ)	
2146	VWAL=CUV*TE(IJ)	
2147	UWAL=CUV*TE(IJ)	
2148	UWAL=CUV*TE(IJ)	
2149	UWAL=CUV*TE(IJ)	
2150	UWAL=CUV*TE(IJ)	
2151	SU(IJ)=GREAT*UWAL	
2152	BP(IJ)=GREAT	
2153	CONTINUE	
2154	AW(IJ)=0.	
2155	C-----EAST BOUNDARY	
2156	DO 1440 J=2,NJM	
2157	IJ=IMNJ(NJM)+J	
2158	GO TO (1441,1442,1442,1443) JTBE(J)	
2159	CONTINUE	
2160	SU(IJ)=SU(IJ)+AE(IJ)*U2(IJ+NJ)	
2161	BP(IJ)=BP(IJ)+AE(IJ)	
2162	GO TO 1442	
2163	CONTINUE	
2164	TE(IJ)=ABS(TE(IJ))	
2165	UWAL=CUV*TE(IJ)	
2166	VWAL=CUV*TE(IJ)	
2167	UWAL=CUV*TE(IJ)	
2168	UWAL=CUV*TE(IJ)	
2169	UWAL=CUV*TE(IJ)	
2170	UWAL=CUV*TE(IJ)	
2171	SU(IJ)=GREAT*UWAL	
2172	BP(IJ)=GREAT	
2173	CONTINUE	
2174	AE(IJ)=0.	
2175	C	
2176	RETURN	
2177	C	
2178	C-----BOUNDARY CONDITIONS FOR V2-REYNOLDS STRESS	
2179	C	
2180	CONTINUE	
2181	C-----SOUTH BOUNDARY	
2182	DO 1510 I=2,NIM	
2183	IJ=IMNJ(I)+2	
2184	GO TO (1511,1512,1512,1513) ITBS(I)	
2185	CONTINUE	
2186	SU(IJ)=SU(IJ)+AS(IJ)*V2(IJ-1)	
2187	BP(IJ)=BP(IJ)+AS(IJ)	
2188	GO TO 1512	
2189	CONTINUE	
2190	TE(IJ)=ABS(TE(IJ))	
2191	UWAL=CUV*TE(IJ)	
2192	VWAL=CUV*TE(IJ)	
2193	UWAL=CUV*TE(IJ)	
2194	UWAL=CUV*TE(IJ)	
2195	UWAL=CUV*TE(IJ)	
2196	UWAL=CUV*TE(IJ)	
2197	SU(IJ)=GREAT*UWAL	
2198	BP(IJ)=GREAT	
2199	CONTINUE	
2200	AS(IJ)=0.	
2201	C-----NORTH BOUNDARY	

Oct 12 1996 16:39	rsmod_2d	Page 32
2202	DO 1520 I=2,NIM	
2203	IJ=IMNJ(I)+NJM	
2204	GO TO (1521,1522,1522,1523) ITBN(I)	
2205	CONTINUE	
2206	SU(IJ)=SU(IJ)+AN(IJ)*V2(IJ+1)	
2207	BP(IJ)=BP(IJ)+AN(IJ)	
2208	GO TO 1522	
2209	CONTINUE	
2210	TE(IJ)=ABS(TE(IJ))	
2211	UWAL=CUV*TE(IJ)	
2212	VWAL=CUV*TE(IJ)	
2213	UWAL=CUV*TE(IJ)	
2214	UWAL=CUV*TE(IJ)	
2215	UWAL=CUV*TE(IJ)	
2216	UWAL=CUV*TE(IJ)	
2217	UWAL=CUV*TE(IJ)	
2218	UWAL=CUV*TE(IJ)	
2219	SU(IJ)=GREAT*UWAL	
2220	BP(IJ)=GREAT	
2221	CONTINUE	
2222	AN(IJ)=0.	
2223	C-----WEST BOUNDARY	
2224	DO 1530 J=2,NJM	
2225	IJ=IMNJ(2)+J	
2226	GO TO (1531,1532,1532,1533) JTBW(J)	
2227	CONTINUE	
2228	SU(IJ)=SU(IJ)+AW(IJ)*V2(IJ-NJ)	
2229	BP(IJ)=BP(IJ)+AW(IJ)	
2230	GO TO 1532	
2231	CONTINUE	
2232	TE(IJ)=ABS(TE(IJ))	
2233	UWAL=CUV*TE(IJ)	
2234	VWAL=CUV*TE(IJ)	
2235	UWAL=CUV*TE(IJ)	
2236	UWAL=CUV*TE(IJ)	
2237	UWAL=CUV*TE(IJ)	
2238	UWAL=CUV*TE(IJ)	
2239	SU(IJ)=GREAT*UWAL	
2240	BP(IJ)=GREAT	
2241	CONTINUE	
2242	AW(IJ)=0.	
2243	C-----EAST BOUNDARY	
2244	DO 1540 J=2,NJM	
2245	IJ=IMNJ(NJM)+J	
2246	GO TO (1541,1542,1542,1543) JTBE(J)	
2247	CONTINUE	
2248	SU(IJ)=SU(IJ)+AE(IJ)*V2(IJ+NJ)	
2249	BP(IJ)=BP(IJ)+AE(IJ)	
2250	GO TO 1542	
2251	CONTINUE	
2252	TE(IJ)=ABS(TE(IJ))	
2253	UWAL=CUV*TE(IJ)	
2254	VWAL=CUV*TE(IJ)	
2255	UWAL=CUV*TE(IJ)	
2256	UWAL=CUV*TE(IJ)	
2257	UWAL=CUV*TE(IJ)	
2258	SU(IJ)=GREAT*UWAL	
2259	BP(IJ)=GREAT	
2260	CONTINUE	
2261	AE(IJ)=0.	
2262	C	
2263	RETURN	
2264	C	
2265	C-----BOUNDARY CONDITIONS FOR W2-REYNOLDS STRESS	
2266	C	
2267	CONTINUE	
2268	C-----SOUTH BOUNDARY	
2269	DO 1610 I=2,NIM	
2270	IJ=IMNJ(I)+2	
2271	GO TO (1611,1612,1612,1613) ITBS(I)	
2272	CONTINUE	
2273	SU(IJ)=SU(IJ)+AS(IJ)*W2(IJ-1)	

Oct 12 1996 16:39	rsmmod_2d	Page 34
2344	ARW= SORT (DXB**2+DYB**2)*SMALL	
2345	DXB=DXB/ARW	
2346	DYB=DYB/ARW	
2347	VP2=U(IJ)*DXB+V(IJ)*DYB	
2348	GRADVP2=ABS(VP2)-UWALL	
2349	SIGN=1.0	
2350	IF (GRADVP2.LT.0.) SIGN=-1.0	
2351	TE(IJ)=ABS(TE(IJ))	
2352	UWAL=CUV*TE(IJ)	
2353	VWAL=CUV*TE(IJ)	
2354	UVWAL=CUV*TE(IJ)	
2355	UVREAL=UWAL*FTLIS(I)*FT2S(I)+VWAL*FNLIS(I)*FN2S(I)+	
2356	& UVWAL*(FTLIS(I)*FN2S(I)+FT2S(I)*FNLIS(I))	
2357	SU(IJ)=GREAT*SIGN*UVREAL	
2358	BP(IJ)=GREAT	
2359	CONTINUE	
2360	AS(IJ)=0.	
2361	1710 CONTINUE	
2362	C-----NORTH BOUNDARY	
2363	DO 1720 I=2,NJM	
2364	IJ=IMNJ(I)+NJM	
2365	GO TO (1721,1722,1722,1723) ITBN(I)	
2366	CONTINUE	
2367	SU(IJ)=SU(IJ)+AN(IJ)*UV(IJ+1)	
2368	BP(IJ)=BP(IJ)+AN(IJ)	
2369	GO TO 1722	
2370	CONTINUE	
2371	DXB=XX(IJ)-XX(IJ-NJ)	
2372	DYB=YY(IJ)-YY(IJ-NJ)	
2373	ARW= SORT (DXB**2+DYB**2)*SMALL	
2374	DXB=DXB/ARW	
2375	DYB=DYB/ARW	
2376	VP2=U(IJ)*DXB+V(IJ)*DYB	
2377	GRADVP2=UWALL-ABS(VP2)	
2378	SIGN=1.0	
2379	IF (GRADVP2.LT.0.) SIGN=-1.0	
2380	TE(IJ)=ABS(TE(IJ))	
2381	UWAL=CUV*TE(IJ)	
2382	VWAL=CUV*TE(IJ)	
2383	UVWAL=CUV*TE(IJ)	
2384	UVREAL=UWAL*FTLIS(I)*FT2N(I)+VWAL*FNLIS(I)*FN2N(I)+	
2385	& UVWAL*(FTLIS(I)*FN2N(I)+FT2N(I)*FNLIS(I))	
2386	SU(IJ)=GREAT*SIGN*UVREAL	
2387	BP(IJ)=GREAT	
2388	CONTINUE	
2389	AN(IJ)=0.	
2390	CONTINUE	
2391	C-----WEST BOUNDARY	
2392	DO 1730 J=2,NJM	
2393	IJ=IMNJ(2)+J	
2394	GO TO (1731,1732,1732,1733) JTBN(J)	
2395	CONTINUE	
2396	SU(IJ)=SU(IJ)+AW(IJ)*UV(IJ-NJ)	
2397	BP(IJ)=BP(IJ)+AW(IJ)	
2398	GO TO 1732	
2399	CONTINUE	
2400	DXB=XX(IJ-NJ)-XX(IJ-NJ-1)	
2401	DYB=YY(IJ-NJ)-YY(IJ-NJ-1)	
2402	ARW= SORT (DXB**2+DYB**2)*SMALL	
2403	DXB=DXB/ARW	
2404	DYB=DYB/ARW	
2405	VP2=U(IJ)*DXB+V(IJ)*DYB	
2406	GRADVP2=ABS(VP2)-VWALL	
2407	SIGN=1.0	
2408	IF (GRADVP2.LT.0.) SIGN=-1.0	
2409	TE(IJ)=ABS(TE(IJ))	
2410	UWAL=CUV*TE(IJ)	
2411	VWAL=CUV*TE(IJ)	
2412	UVWAL=CUV*TE(IJ)	
2413	UVREAL=UWAL*FTLIS(I)*FT2W(J)+VWAL*FNLIS(I)*FN2W(J)+	
2414	& UVWAL*(FTLIS(I)*FN2W(J)+FT2W(J)*FNLIS(I))	

Oct 12 1996 16:39	rsmmod_2d	Page 33
2273	BP(IJ)=BP(IJ)+AS(IJ)	
2274	GO TO 1612	
2275	CONTINUE	
2276	TE(IJ)=ABS(TE(IJ))	
2277	SU(IJ)=GREAT*CWV*TE(IJ)	
2278	BP(IJ)=GREAT	
2279	CONTINUE	
2280	AS(IJ)=0.	
2281	1610 CONTINUE	
2282	C-----NORTH BOUNDARY	
2283	DO 1620 I=2,NJM	
2284	IJ=IMNJ(I)+NJM	
2285	GO TO (1621,1622,1622,1623) ITBN(I)	
2286	CONTINUE	
2287	SU(IJ)=SU(IJ)+AN(IJ)*W2(IJ+1)	
2288	BP(IJ)=BP(IJ)+AN(IJ)	
2289	GO TO 1622	
2290	CONTINUE	
2291	TE(IJ)=ABS(TE(IJ))	
2292	SU(IJ)=GREAT*CWV*TE(IJ)	
2293	BP(IJ)=GREAT	
2294	CONTINUE	
2295	AN(IJ)=0.	
2296	1620 CONTINUE	
2297	C-----WEST BOUNDARY	
2298	DO 1630 J=2,NJM	
2299	IJ=IMNJ(2)+J	
2300	GO TO (1631,1632,1632,1633) JTBN(J)	
2301	CONTINUE	
2302	SU(IJ)=SU(IJ)+AW(IJ)*W2(IJ-NJ)	
2303	BP(IJ)=BP(IJ)+AW(IJ)	
2304	GO TO 1632	
2305	CONTINUE	
2306	TE(IJ)=ABS(TE(IJ))	
2307	SU(IJ)=GREAT*CWV*TE(IJ)	
2308	BP(IJ)=GREAT	
2309	CONTINUE	
2310	AW(IJ)=0.	
2311	1630 CONTINUE	
2312	C-----EAST BOUNDARY	
2313	DO 1640 J=2,NJM	
2314	IJ=IMNJ(NJM)+J	
2315	GO TO (1641,1642,1642,1643) JTBN(J)	
2316	CONTINUE	
2317	SU(IJ)=SU(IJ)+AE(IJ)*W2(IJ+NJ)	
2318	BP(IJ)=BP(IJ)+AE(IJ)	
2319	GO TO 1642	
2320	CONTINUE	
2321	TE(IJ)=ABS(TE(IJ))	
2322	SU(IJ)=GREAT*CWV*TE(IJ)	
2323	BP(IJ)=GREAT	
2324	CONTINUE	
2325	AE(IJ)=0.	
2326	1640 CONTINUE	
2327	C	
2328	RETURN	
2329	C	
2330	C-----BOUNDARY CONDITIONS FOR UV-REYNOLDS STRESS	
2331	C	
2332	CONTINUE	
2333	C-----SOUTH BOUNDARY	
2334	DO 1710 I=2,NJM	
2335	IJ=IMNJ(I)+2	
2336	GO TO (1711,1712,1712,1713) ITBN(I)	
2337	CONTINUE	
2338	SU(IJ)=SU(IJ)+AS(IJ)*UV(IJ-1)	
2339	BP(IJ)=BP(IJ)+AS(IJ)	
2340	GO TO 1712	
2341	CONTINUE	
2342	DXB=XX(IJ-1)-XX(IJ-NJ-1)	
2343	DYB=YY(IJ-1)-YY(IJ-NJ-1)	

```

2415 SU(IJ)=GREAT*SIGN*UVREAL
2416 BP(IJ)=GREAT
2417 1732 CONTINUE
2418 AW(IJ)=0.
2419 1730 CONTINUE
2420 C-----EAST BOUNDARY
2421 DO 1740 J=2,NJM
2422 IJ=IMNJ(NIM)+J
2423 GO TO (1741,1742,1743) JTBE(J)
2424 1741 CONTINUE
2425 SU(IJ)=SU(IJ)+AE(IJ)*UV(IJ+NJ)
2426 BP(IJ)=BP(IJ)+AE(IJ)
2427 GO TO 1742
2428 1743 CONTINUE
2429 DXB=XX(IJ)-XX(IJ-1)
2430 DYB=YY(IJ)-YY(IJ-1)
2431 ARB=SQRT(DXB**2+DYB**2)+SMALL
2432 DXB=DXB/ARB
2433 DYB=DYB/ARB
2434 VP2=U(IJ)*DXB+V(IJ)*DYB
2435 GRADVP2=VWALL-ABS(VP2)
2436 SIGN=1.0
2437 IF (GRADVP2.LT.0.) SIGN=-1.0
2438 TE(IJ)=ABS(TE(IJ))
2439 UWAL=CWV*TE(IJ)
2440 VWAL=CVW*TE(IJ)
2441 UVREAL=UWAL*FT1E(IJ)
2442 & UVWAL*(FT1E(IJ)*FN2E(J)+FT2E(J)*FN1E(J))
2443 SU(IJ)=GREAT*SIGN*UVREAL
2444 BP(IJ)=GREAT
2445 1742 CONTINUE
2446 AE(IJ)=0.
2447 1740 CONTINUE
2448 C
2449 C
2450 C
2451 C
2452 C-----BOUNDARY CONDITIONS FOR VW-REYNOLDS STRESS
2453 C
2454 1800 CONTINUE
2455 C-----SOUTH BOUNDARY
2456 DO 1810 I=2,NIM
2457 IJ=IMNJ(I)+2
2458 GO TO (1811,1812,1813) ITBS(I)
2459 1811 CONTINUE
2460 SU(IJ)=SU(IJ)+AS(IJ)*VW(IJ-1)
2461 BP(IJ)=BP(IJ)+AS(IJ)
2462 GO TO 1812
2463 1813 CONTINUE
2464 TE(IJ)=ABS(TE(IJ))
2465 UWAL=CWV*TE(IJ)
2466 VWAL=CVW*TE(IJ)
2467 VWREAL=UWAL*FT2S(I)+VWAL*FN2S(I)
2468 SU(IJ)=GREAT*VWREAL
2469 BP(IJ)=GREAT
2470 1812 CONTINUE
2471 AS(IJ)=0.
2472 1810 CONTINUE
2473 C-----NORTH BOUNDARY
2474 DO 1820 I=2,NIM
2475 IJ=IMNJ(I)+NJM
2476 GO TO (1821,1822,1823) ITBN(I)
2477 1821 CONTINUE
2478 SU(IJ)=SU(IJ)+AN(IJ)*VW(IJ+1)
2479 BP(IJ)=BP(IJ)+AN(IJ)
2480 GO TO 1822
2481 1823 CONTINUE
2482 TE(IJ)=ABS(TE(IJ))
2483 UWAL=CVW*TE(IJ)
2484 VWAL=CVW*TE(IJ)
2485 VWREAL=UWAL*FT2N(I)+VWAL*FN2N(I)

```

```

2486 SU(IJ)=GREAT*VWREAL
2487 BP(IJ)=GREAT
2488 1822 CONTINUE
2489 AN(IJ)=0.
2490 1820 CONTINUE
2491 C-----WEST BOUNDARY
2492 DO 1830 J=2,NJM
2493 IJ=IMNJ(2)+J
2494 GO TO (1831,1832,1833) JTBW(J)
2495 1831 CONTINUE
2496 SU(IJ)=SU(IJ)+AW(IJ)*VW(IJ-NJ)
2497 BP(IJ)=BP(IJ)+AW(IJ)
2498 GO TO 1832
2499 1833 CONTINUE
2500 TE(IJ)=ABS(TE(IJ))
2501 UWAL=CVW*TE(IJ)
2502 VWAL=CVW*TE(IJ)
2503 VWREAL=UWAL*FT2W(J)+VWAL*FN2W(J)
2504 SU(IJ)=GREAT*VWREAL
2505 BP(IJ)=GREAT
2506 1832 CONTINUE
2507 AW(IJ)=0.
2508 1830 CONTINUE
2509 C-----EAST BOUNDARY
2510 DO 1840 J=2,NJM
2511 IJ=IMNJ(NIM)+J
2512 GO TO (1841,1842,1843) JTBE(J)
2513 1841 CONTINUE
2514 SU(IJ)=SU(IJ)+AE(IJ)*VW(IJ+NJ)
2515 BP(IJ)=BP(IJ)+AE(IJ)
2516 GO TO 1842
2517 1843 CONTINUE
2518 TE(IJ)=ABS(TE(IJ))
2519 UWAL=CVW*TE(IJ)
2520 VWAL=CVW*TE(IJ)
2521 VWREAL=UWAL*FT2E(J)+VWAL*FN2E(J)
2522 SU(IJ)=GREAT*VWREAL
2523 BP(IJ)=GREAT
2524 1842 CONTINUE
2525 AE(IJ)=0.
2526 1840 CONTINUE
2527 C
2528 C
2529 C
2530 C-----BOUNDARY CONDITIONS FOR UW-REYNOLDS STRESS
2531 C
2532 1900 CONTINUE
2533 C-----SOUTH BOUNDARY
2534 DO 1910 I=2,NIM
2535 IJ=IMNJ(I)+2
2536 GO TO (1911,1912,1913) ITBS(I)
2537 1911 CONTINUE
2538 SU(IJ)=SU(IJ)+AS(IJ)*UW(IJ-1)
2539 BP(IJ)=BP(IJ)+AS(IJ)
2540 GO TO 1912
2541 1913 CONTINUE
2542 TE(IJ)=ABS(TE(IJ))
2543 UWAL=CVW*TE(IJ)
2544 VWAL=CVW*TE(IJ)
2545 UWREAL=UWAL*FT1S(I)+VWAL*FN1S(I)
2546 SU(IJ)=GREAT*UWREAL
2547 BP(IJ)=GREAT
2548 1912 CONTINUE
2549 AS(IJ)=0.
2550 1910 CONTINUE
2551 C-----NORTH BOUNDARY
2552 DO 1920 I=2,NIM
2553 IJ=IMNJ(I)+NJM
2554 GO TO (1921,1922,1923) ITBN(I)
2555 1921 CONTINUE
2556 SU(IJ)=SU(IJ)+AN(IJ)*UW(IJ+1)

```

Oct 12 1996 16:39	rsmod_2d	Page 37
2557	BP(IJ)=BP(IJ)+AN(IJ)	
2558	GO TO 1922	
2559	CONTINUE	
2560	TE(IJ)=ABS(TE(IJ))	
2561	UWAL=CUM*TE(IJ)	
2562	UWAL=UWAL*FTIN(I)+VWAL*FNIN(I)	
2563	UWREAL=UWAL*UWREAL	
2564	BP(IJ)=GREAT*UWREAL	
2565	BP(IJ)=GREAT	
2566	CONTINUE	
2567	AN(IJ)=0.	
2568	CONTINUE	
2569	C-----WEST BOUNDARY	
2570	DO 1930 J=2,NJM	
2571	IJ=IMNJ(2)+J	
2572	GO TO (1931,1932,1933) JTBW(J)	
2573	CONTINUE	
2574	SU(IJ)=SU(IJ)+AW(IJ)*UW(IJ-NJ)	
2575	BP(IJ)=BP(IJ)+AW(IJ)	
2576	GO TO 1932	
2577	CONTINUE	
2578	TE(IJ)=ABS(TE(IJ))	
2579	UWAL=CUM*TE(IJ)	
2580	UWAL=UWAL*FTIN(J)+VWAL*FNIN(J)	
2581	UWREAL=UWAL*UWREAL	
2582	SU(IJ)=GREAT*UWREAL	
2583	BP(IJ)=GREAT	
2584	CONTINUE	
2585	AW(IJ)=0.	
2586	CONTINUE	
2587	C-----EAST BOUNDARY	
2588	DO 1940 J=2,NJM	
2589	IJ=IMNJ(NIM)+J	
2590	GO TO (1941,1942,1943) JTBE(J)	
2591	CONTINUE	
2592	SU(IJ)=SU(IJ)+AE(IJ)*UW(IJ+NJ)	
2593	BP(IJ)=BP(IJ)+AE(IJ)	
2594	GO TO 1942	
2595	CONTINUE	
2596	TE(IJ)=ABS(TE(IJ))	
2597	UWAL=CUM*TE(IJ)	
2598	UWAL=UWAL*FTIE(J)+VWAL*FNIE(J)	
2599	UWREAL=UWAL*UWREAL	
2600	SU(IJ)=GREAT*UWREAL	
2601	BP(IJ)=GREAT	
2602	CONTINUE	
2603	AE(IJ)=0.	
2604	CONTINUE	
2605	C	
2606	RETURN	
2607	END	
2608	C*****	
2609	SUBROUTINE ASOLSIP(PHI,IPHI,RESOR)	
2610	C*****	
2611	C*****	
2612	C	
2613	C	
2614	C	
2615	C	
2616	C	
2617	C	
2618	DIMENSION PHI(NXNY), RES(NXNY)	
2619	DIMENSION BS(NXNY), BN(NXNY), BE(NXNY), BW(NXNY)	
2620	DIMENSION FP(NXNY), RESOR(2)	
2621	C	
2622	NJ = NJM + 1	
2623	NI = NJM + 1	
2624	NINJ = NI*NJ	
2625	DO 5 IJ=1,NINJ	
2626	BN(IJ) = 0.	
2627	BE(IJ) = 0.	

Oct 12 1996 16:39	rsmod_2d	Page 38
2628	RES(IJ) = 1.	
2629	FP(IJ)=PHI(IJ)	
2630	5 CONTINUE	
2631	C	
2632	C	
2633	C	
2634	DO 10 I=2,NIM	
2635	DO 10 J=2,NJM	
2636	IJ=IMNJ(I)+J	
2637	API=1.0/AP(IJ)	
2638	AP(IJ)=1.0	
2639	AE(IJ)=AE(IJ)*API	
2640	AN(IJ)=AN(IJ)*API	
2641	AW(IJ)=AW(IJ)*API	
2642	AS(IJ)=AS(IJ)*API	
2643	SU(IJ)=SU(IJ)*API	
2644	10 CONTINUE	
2645	C	
2646	DO 20 I=2,NIM	
2647	DO 20 J=2,NJM	
2648	IJ=IMNJ(I)+J	
2649	IGN=IJ-1	
2650	IMJ=IJ-NJ	
2651	BW(IJ)=-AW(IJ)/(1.+ALFAKE*BN(IJ-NJ))	
2652	BS(IJ)=-AS(IJ)/(1.+ALFAKE*BE(IJ))	
2653	POM1=ALFAKE*BW(IJ)*BN(IMJ)	
2654	POM2=ALFAKE*BS(IJ)*BE(IJM)	
2655	BP(IJ)=AP(IJ)+POM1+POM2-BW(IJ)*BE(IMJ)-BS(IJ)*BN(IJM)	
2656	BN(IJ)=-AN(IJ)-POM1/(BP(IJ)+SMALL)	
2657	BE(IJ)=-AE(IJ)-POM2/(BP(IJ)+SMALL)	
2658	20 CONTINUE	
2659	C	
2660	DO 100 L=1,NSWPKE(IPHI)	
2661	RESORP=0.	
2662	DO 30 I=2,NIM	
2663	DO 30 J=2,NJM	
2664	IJ=IMNJ(I)+J	
2665	RES(IJ)=AN(IJ)*PHI(IJ-NJ)+SU(IJ)-AP(IJ)*PHI(IJ)	
2666	& AW(IJ)*PHI(IJ-NJ)+SU(IJ)-AP(IJ)*PHI(IJ)	
2667	RESORP=RESORP+ABS(RES(IJ))	
2668	RES(IJ)=(RES(IJ)-BS(IJ)*RES(IJ-1)-BW(IJ)*RES(IJ-NJ))/	
2669	(BP(IJ)+SMALL)	
2670	30 CONTINUE	
2671	C	
2672	IF(L.EQ.1) RESORKE(IPHI)=RESORP	
2673	RSM=SORKE(IPHI)*RESORKE(IPHI)	
2674	DO 40 I=2,NIM	
2675	II=NIM+2-I	
2676	DO 40 J=2,NJM	
2677	JJ=NJM+2-J	
2678	IJ=IMNJ(II)+JJ	
2679	RES(IJ)=RES(IJ)-BN(IJ)*RES(IJ+1)-BE(IJ)*RES(IJ+NJ)	
2680	PHI(IJ)=PHI(IJ)+RES(IJ)	
2681	40 CONTINUE	
2682	IF(RESORP.LE.RSM) RETURN	
2683	IF(RESORP.LE.RSM) GOTO 200	
2684	100 CONTINUE	
2685	C	
2686	IF(RESORP.GE.RSM.AND.L.GE.NSWPKE(IPHI)) WRITE(*,2)	
2687	2 FORMAT(10X,' SOLSIP DID NOT CONVERGE '	
2688	C	
2689	200 CONTINUE	
2690	AUX1=0.	
2691	AUX2=0.	
2692	DO 50 I=2,NIM	
2693	DO 50 J=2,NJM	
2694	IJ=IMNJ(I)+J	
2695	AUX1=AUX1+ABS(PHI(IJ)-FP(IJ))	
2696	AUX2=AUX2+ABS(FP(IJ))	
2697	50 CONTINUE	
2698	C	
2699	IF(AUX2.LT.1.E-30) AUX2=1.E-30	

Oct 12 1996 16:39	rsmod_2d	Page 39
2699	RESOR(IPHI)=AUX1/AUX2	
2700	RETURN	
2701	END	
2702	C	
2703	C.....A GAUSS ELIMINATION SOLVER	
2704	SUBROUTINE SOLV(A,BB,N)	
2705	DIMENSION A(N,N),B(N),C(N),BB(N),X(N),MME(N)	
2706	EP=1.E-19	
2707	DO 10 J=1,N	
2708	MME(J)=J	
2709	DO 20 I=1,N	
2710	Y=0.	
2711	DO 30 J=I,N	
2712	IF(ABS(A(I,J)).LT.ABS(Y)) GOTO 30	
2713	K=J	
2714	Y=A(I,J)	
2715	CONTINUE	
2716	C	
2717	IF(ABS(Y).LT.EP) THEN	
2718	WRITE(*,*)	
2719	WRITE(9,*)	
2720	DO 35 IA=1,N	
2721	WRITE(*,1000) (A(IA,JA),JA=1,N)	
2722	CONTINUE	
2723	PRINT*,'THERE IS NO CONVERSE MATRIX'	
2724	STOP 2222	
2725	ENDIF	
2726	C	
2727	Y=1./Y	
2728	DO 40 J=1,N	
2729	C(J)=A(J,K)	
2730	A(J,K)=A(J,I)	
2731	A(J,I)=-C(J)*Y	
2732	B(J)=A(I,J)*Y	
2733	A(I,J)=A(I,J)*Y	
2734	A(I,I)=Y	
2735	J=MME(I)	
2736	MME(I)=MME(K)	
2737	MME(K)=J	
2738	DO 11 K=1,N	
2739	IF(K.EQ.I) GOTO 11	
2740	DO 12 J=1,N	
2741	IF(J.EQ.I) GOTO 12	
2742	A(K,J)=A(K,J)-B(J)*C(K)	
2743	CONTINUE	
2744	CONTINUE	
2745	CONTINUE	
2746	DO 33 I=1,N	
2747	DO 44 K=1,N	
2748	IF(MME(K).EQ.I) GOTO 55	
2749	CONTINUE	
2750	IF(K.EQ.I) GOTO 33	
2751	DO 66 J=1,N	
2752	W=A(I,J)	
2753	A(I,J)=A(K,J)	
2754	A(K,J)=W	
2755	IW=MME(I)	
2756	MME(I)=MME(K)	
2757	MME(K)=IW	
2758	CONTINUE	
2759	1000 FORMAT(4X,1P5E13.4)	
2760	DO 50 I=1,N	
2761	X(I)=0.	
2762	DO 50 J=1,N	
2763	X(I)=X(I)+A(I,J)*BB(J)	
2764	DO 60 I=1,N	
2765	BB(I)=X(I)	
2766	RETURN	
2767	END	
2768	C-----SUBROUTINE AMODIFY(SUASM,SVASM,SWASM,	
2769		

Oct 12 1996 16:39	rsmod_2d	Page 40
2770	& X,Y,FX,FY,ARE,VOL,R,ICAL,AKSI,DEN,	
2771	& VISCOS,TE,DNS,DNN,DNW,DNE,ITBS,ITBN,ITBW,ITBE,	
2772	& U,V,W,ITER)	
2773	C-----	
2774	INCLUDE 'gridparam.h'	
2775	INCLUDE 'rsm.h'	
2776	C	
2777	DIMENSION UDW(NY),VWV(NY),UVW(NY),UWV(NY),VWV(NY)	
2778	DIMENSION X(NXNY),Y(NXNY),FX(NXNY),FY(NXNY),R(NXNY)	
2779	DIMENSION DNS(NX),DNN(NX),DNW(NX),DNE(NX)	
2780	DIMENSION U(NXNY),V(NXNY),W(NXNY),DEN(NXNY),TE(NXNY)	
2781	DIMENSION ITBS(NX),ITBN(NX),ITBW(NX),ITBE(NX)	
2782	IF(ITER.EQ.1) THEN	
2783	REWIN 41	
2784	READ(41,*)	
2785	READ(41,*)	
2786	READ(41,*) CD1,CD2,CMU,ELOG,CAPPA	
2787	READ(41,*)	
2788	READ(41,*) LRE,LAY2	
2789	END IF	
2790	C	
2791	NI = NIM + 1	
2792	NJ = NJM + 1	
2793	CMU25=SQRT(SQRT(CMU))	
2794	C	
2795	C-----SOUTH BOUNDARY	
2796	DO 600 I=2,NIM	
2797	IJ=IMNJ(I)+2	
2798	IF(ITBS(I).EQ.4) THEN	
2799	DXB=X(IJ-1)-X(IJ-NJ-1)	
2800	DYB=Y(IJ-1)-Y(IJ-NJ-1)	
2801	ARW=SQRT(DXB**2+DYB**2)	
2802	DXB=DXB/ARW	
2803	DYB=DYB/ARW	
2804	CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))	
2805	YPLS=DNS(I)*CONST/VISCOS	
2806	IF (YPLS.LE.11.63.OR.LAY2) THEN	
2807	TCOEF=VISCOS/DNS(I)	
2808	ELSE	
2809	UPLUS=LOG(ELOG*YPLS)/CAPPA	
2810	TCOEF=CONST/UPLUS	
2811	ENDIF	
2812	VPINT=U(IJ)*DXB+V(IJ)*DYB	
2813	VPINT=VPINT+W(IJ)	
2814	VPINT=ABS(VPINT-SQRT(U(IJ-1)*U(IJ-1)+	
2815	V(IJ-1)*V(IJ-1)+W(IJ-1)*W(IJ-1)))	
2816	1 GENTS(I)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNS(I))	
2817	C	
2818	ENDIF	
2819	C-----NORTH BOUNDARY	
2820	IJ=IMNJ(I)+NMJ	
2821	IF (ITBN(I).EQ.4) THEN	
2822	DXB=X(IJ)-X(IJ-NJ)	
2823	DYB=Y(IJ)-Y(IJ-NJ)	
2824	ARW=SQRT(DXB**2+DYB**2)	
2825	DXB=DXB/ARW	
2826	DYB=DYB/ARW	
2827	CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))	
2828	YPLS=DNN(I)*CONST/VISCOS	
2829	IF (YPLS.LE.11.63.OR.LAY2) THEN	
2830	TCOEF=VISCOS/DNN(I)	
2831	ELSE	
2832	UPLUS=LOG(ELOG*YPLS)/CAPPA	
2833	TCOEF=CONST/UPLUS	
2834	ENDIF	
2835	VPINT=U(IJ)*DXB+V(IJ)*DYB	
2836	VPINT=VPINT+W(IJ)	
2837	VPINT=ABS(VPINT-SQRT(U(IJ+1)*U(IJ+1)+	
2838	V(IJ+1)*V(IJ+1)+W(IJ+1)*W(IJ+1)))	
2839	1 GENTN(I)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNN(I))	
2840	C	

Oct 12 1996 16:39

rsmmod_2d

Page 41

```

2841      ENDIF
2842      600 CONTINUE
2843      C-----WEST BOUNDARY
2844      DO 620 J=2,NJM
2845          IJ=IMNJ(2)+J
2846          IF (JTBW(J).EQ.4) THEN
2847              DXB=X(IJ-NJ)-X(IJ-NJ-1)
2848              DYB=Y(IJ-NJ)-Y(IJ-NJ-1)
2849              ARW=SQRT(DXB**2+DYB**2)
2850              DXB=DXB/ARW
2851              DYB=DYB/ARW
2852              CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
2853              YPLS=DNW(J)*CONST/VISCOS
2854              IF (YPLS.LE.11.63.OR.LAY2) THEN
2855                  TCOEF=VISCOS/DNW(J)
2856              ELSE
2857                  UPLS=LOG(ELOG*YPLS)/CAPPA
2858                  TCOEF=CONST/UPLS
2859              ENDIF
2860              VPINT=U(IJ)*DXB+V(IJ)*DYB
2861              VPINT=VPINT*W(IJ)
2862              VPINT=ABS(VPINT-SQRT(U(IJ-NJ)*U(IJ-NJ)+
2863                  V(IJ-NJ)*V(IJ-NJ)+W(IJ-NJ)*W(IJ-NJ)))
2864              1 GENTW(J)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNW(J))
2865      ENDIF
2866      C-----EAST BOUNDARY
2867      IJ=IMNJ(NIM)+J
2868      IF (JTBE(J).EQ.4) THEN
2869          UP=U(IJ)
2870          VP=V(IJ)
2871          WP=W(IJ)
2872          UWALL=U(IJ+NJ)
2873          VWALL=V(IJ+NJ)
2874          WWALL=W(IJ+NJ)
2875          TEPR=SQRT(TE(IJ))
2876          DELN=DNE(J)
2877          RB=HAF*(R(IJ)+R(IJ-1))
2878          DENS=DEN(IJ)
2879          DXB=X(IJ)-X(IJ-1)
2880          DYB=Y(IJ)-Y(IJ-1)
2881          ARW=SQRT(DXB**2+DYB**2)
2882          DXB=DXB/ARW
2883          DYB=DYB/ARW
2884          CONST=DEN(IJ)*CMU25*SQRT(TE(IJ))
2885          YPLS=DNE(J)*CONST/VISCOS
2886          IF (YPLS.LE.11.63.OR.LAY2) THEN
2887              TCOEF=VISCOS/DNE(J)
2888          ELSE
2889              UPLS=LOG(ELOG*YPLS)/CAPPA
2890              TCOEF=CONST/UPLS
2891          ENDIF
2892          VPINT=U(IJ)*DXB+V(IJ)*DYB
2893          VPINT=VPINT*W(IJ)
2894          VPINT=ABS(VPINT-SQRT(U(IJ+NJ)*U(IJ+NJ)+
2895              V(IJ+NJ)*V(IJ+NJ)+W(IJ+NJ)*W(IJ+NJ)))
2896          1 GENTEE(J)=TCOEF*CONST*ABS(VPINT)/(CAPPA*DEN(IJ)*DNE(J))
2897      ENDIF
2898      620 CONTINUE
2899      C
2900      RETURN
2901      END
2902      C
2903      C-----
2904      C gridparam.h
2905      C
2906      PARAMETER (HAF=0.5,QTR=0.25)
2907      PARAMETER (SMALL=1.E-30,GREAT=1.E30)
2908      PARAMETER (NX=100)
2909      PARAMETER (NY=50)
2910      PARAMETER (NKNY=NX*NY)
2911      C

```

Oct 12 1996 16:39

rsmmod_2d

Page 42

```

2912      C-----
2913      C rsm.h
2914      PARAMETER (HAF=0.5,QTR=0.25)
2915      PARAMETER (SMALL=1.E-30,GREAT=1.E+30)
2916      C
2917      COMMON /A1/ AS(NKNY), AN(NKNY), AE(NKNY), AW(NKNY),
2918          1 AP(NKNY), BP(NKNY), SU(NKNY), AFU(NKNY), APV(NKNY)
2919      C
2920      COMMON /A2/ SORKE(2), NSWPKE(2), URFKE(2), PRTRKE(2)
2921      C
2922      COMMON /A3/ GKE,ALFAKE,RESORKE(2),URFVIS
2923      COMMON /CONST/ CMU,CMU75,ELOG,CAPPA,
2924          1 C1,C2,C1P,C2P,CK,CE,CD1,CD2
2925      COMMON /A3/ GENTS(NX),GENTN(NX),GENTW(NY),GENTEE(NY),P11(NKNY),
2926          1 P22(NKNY),P33(NKNY),P12(NKNY),P13(NKNY),P23(NKNY)
2927      COMMON /A4/ FUNK(NKNY),FUNK(NKNY),FUNKY(NKNY),
2928          1 GEN(NKNY),DUDX(NKNY),DUDY(NKNY),DUDX(NKNY),
2929          2 DUDY(NKNY),DWDX(NKNY),DWDY(NKNY)
2930      COMMON /STRESS/ U2(NKNY),V2(NKNY),W2(NKNY),
2931          1 UV(NKNY),VW(NKNY),OW(NKNY)
2932      C
2933      LOGICAL AKSI,RESTART
2934

```

CHAPTER 6

3D k - ε Turbulence Model

Table of Contents

	page
6.1 Introduction	81
6.2 Theory and Model Equations	81
6.3 Module Evaluation	82
6.4 References	82
Figures	
Appendix E	83

6.1 Introduction

In this section a description of the standard k - ϵ turbulence model that is coded as a self contained computer program to compute turbulent flow quantities in three-dimensional, body-fitted geometry is given. Module structure and variables used are given in the Appendix. The module was successfully tested as a self-contained unit using the REACT code[1].

6.2 Theory and Model Equations

The k - ϵ turbulence model used is based on the standard two equation k - ϵ model of Launder and Spalding [2]. For a steady, incompressible flow the transport equations for the turbulent kinetic energy k and energy dissipation ϵ can be written in generalized Cartesian coordinates as;

$$\frac{\partial \rho U_j k}{\partial x_j} = \frac{1}{r} \frac{\partial}{\partial x_j} \left(\mu + \frac{\mu_t}{\sigma_k} \frac{\partial k}{\partial x_j} \right) + G - \rho \epsilon \quad (1)$$

$$\frac{\partial \rho U_j \epsilon}{\partial x_j} = \frac{1}{r} \frac{\partial}{\partial x_j} \left(\mu + \frac{\mu_t}{\sigma_\epsilon} \frac{\partial \epsilon}{\partial x_j} \right) + G - \rho \epsilon \quad (2)$$

where G denotes the rate of production of turbulent kinetic energy and is expressed as;

$$G = \mu_t \frac{\partial U_i}{\partial x_j} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right)$$

The empirical constants, σ_k , σ_ϵ , C_1 and C_2 have values 1.0, 1.0, 1.44 and 1.92 respectively.

The above equations are valid only in the fully turbulent region away from the wall. Therefore the wall function method (similar to that described in Chapter 2 for the 2D k - ϵ module) is used to model the damping effects of the thin sublayer region close to the wall.

6.3 Module Evaluation

The 3D $k-\epsilon$ turbulent module was evaluated by interfacing the module with the REACT code as the CFD solver and producing the same results that were generated previously with the full REACT code for a centrifugal impeller calculations (Chen et. al [3]). Figure 1 shows the grid topology of the impeller studied with the shroud removed and Figure 2, shows the reduced pressure plot . In general Chen et al's calculations showed good comparisons with experimental data obtained from laser velocimetry in a water test rig.

REFERENCES

- [1] Darian, A. and Chan, D "A User's Manual for the Rocketdyne Elliptic Analysis Code for Turbomachinery" CFD Technology Center, Rocketdyne Division/Rockwell International, 1992.
- [2] Launder, B. E. and Spalding, D. B. "The Numerical Computation of Turbulent Flows" Computer Methods in Applied Mechanics and Engineering, vol. 3, pp. 269-289, 1974.
- [3] Chen, W, Eastland, A., Brozowski, A. and Darian A. "A Comparison of Numerical and Experimental Results of a Centrifugal Impeller Flow" Fifth Int. Symp. on Transport Phenomena and Dynamics of Rotating Machinery, Kaanapali, Hawaii, 1992.
- [4] Stone, H. "Iterative Solution of Implicit Approximations of Multi-Dimensional Partial Differential Equations" SIAM J. Num. Anal., vol. 5, pp. 530 - , 1968.



Figure 1. Impeller grid topology



Figure 2. Reduced pressures

APPENDIX E

3D k - ε Turbulence Module Deck

This module consists of two separate programs KEMOD3 and MODIFY, which have to be linked to the main flow solver. A description of each file will be given next.

Program KEMOD3

This is basically the solver for the k and ε - transport equations. It reads through its argument list different variables from the calling flow solver. These variables are described below.

List of Argument Variable Names

NIM	Number of cell nodes in the I- or ξ -coordinate lines. (input from the flow solver)
NJM	Number of cell nodes in the J- or η -coordinate lines. (input from the flow solver)
NKM	Number of cell nodes in the k- or ζ -coordinate lines. (input from the flow solver)
X	Grid node locations in the x or ξ -direction, dimensioned to X(JXYZ) (JXYZ=NX*NY*NZ) (input from flow solver)
Y	Grid node locations in the y or η -direction, dimensioned to Y(JXYZ) (input from flow solver)
Z	Grid node locations in the z or ζ -direction, dimensioned to Z(JXYZ) (input from flow solver)
U	x-direction velocity (u), dimensioned as U(JXYZ) (input from flow solver)
V	y-direction velocity (v), also dimensional as V(JXYZ) (input from flow solver)
W	z-direction velocity (w), dimensional W(JXYZ) (input from flow solver)
TE	Turbulence kinetic energy k , dimensioned TE(JXYZ) (calculated in the module and returned to the flow solver)

ED	Turbulent energy dissipation rate ϵ , dimensioned ED(JXYZ) (calculated in the module and returned to the flow solver)
URFK	Under-relaxation factor for k -equation (input from flow solver)
URFE	Under-relaxation factor for ϵ -equation (input from flow solver)
PRTK	Prandtl/Schmidt number for turbulent energy-equation, assumed known (input from flow solver)
PRTE	Prandtl/Schmidt number for turbulent energy dissipation equation, assumed known (input from flow solver)
G	= 1.0 if second order upwinding is desired = 0.0 if first order upwinding is used (input from flow solver. Usually calculation of k and ϵ are not very sensitive to the order of upwinding used)
F1	Mass flux variable at cell faces in x - or ξ -direction, dimensioned F1(JXYZ) (input from flow solver)
F2	Mass flux variable at cell faces in y or η -direction, dimensioned F2(JXYZ) (input from flow solver)
F3	Mass flux variable at cell faces in z or ζ -direction, dimensioned F3(JXYZ) (input from flow solver)
ITER	Iteration number (input from flow solver)
VISCOS	Dynamic viscosity (input from flow solver)
VIS	Eddy viscosity, dimensioned VIS(JXYZ) (calculated in the module and returned to the main solver)
C1	Turbulence model constant, C_1 (input from flow solver)
C2	Turbulence model constant, C_2 (input from flow solver)
CMU	Turbulence model constant, C_μ (input from flow solver)
BCFE	Boundary condition flag along east boundary (or y - z plane). It must have one for each boundary node set to: 1-inlet, 2-outlet, 3-symmetry and 4-wall e.g., for an outlet boundary condition on the east boundary set IBCE to $(NY*NZ)*2$, and similarly for other boundaries, dimensioned BCFE(JYZ= $NY*NZ$) (input from flow solver)
BCFW	Boundary condition flag along west boundary, dimensioned BCFW(JYZ) (input from flow solver)
BCFS	Boundary condition flag along the south boundary, dimensioned BCFS(JXZ= $NX*NZ$) (input from flow solver)

BCFN	Boundary condition flag along north boundary, dimensioned BCFN(JXZ) (input from flow solver)
BCFB	Boundary condition flag along bottom boundary (or x-y plane), dimensioned BCFB(JXY=NX*NY) (input from flow solver)
BCFT	Boundary condition flag along top boundary (or x-y plane), dimensioned BCFT(JXY=NX*NY) (input from flow solver)

The module is interfaced with the main flow solver by a call to KEMOD3 with its arguments. For iterative flow solvers the module is called within the iteration sequence after the solution of the momentum equations where the mean velocities are passed to the module. There are different flow solvers utilizing different schemes from staggered to nonstaggered grid arrangement and for nonorthogonal coordinate system there are at least three alternatives to the choice of the velocity components;

- i. Cartesian velocity components
- ii. Contravariant velocity components
- iii. Covariant velocity components

The Cartesian velocity components are the most widely used and have the advantage of simple formulation of the governing equations. Whatever the arrangement used, mass fluxes at cell faces are required and passed to the module as F1, F2 and F3 in all directions. The location of other variables such as k and ε are at the cell center or cell nodes.

The module starts by reassigning variable names passed to it from flow solver to names that are shared with the different subroutines of the module in a common statement file "kemod.h". Then a check is made if it is the first iteration in which case the grid file "GRIDG" is called -after passing the grid node locations X, Y and Z- in order to calculate grid related quantities which will be explained later. The need to call GRIDG can be waived if all the grid data are passed to the module. That is all the information about the grid such as interpolation factors FX, FY and FZ, cell volumes (VOL) and normal distances of first grid point from grid boundaries (DNS from south boundary, DNN - from north boundary, DNW - from west boundary, DNE - from east boundary, DNB - from bottom boundary and DNT - from top boundary).

After this a call to subroutine CALCE is made to calculate the turbulent kinetic energy k (with the identifier IPHI=1). The energy dissipation equation is solved next by a call to subroutine CALCE

again with the identifier IPHI=2. The turbulent viscosity is updated next by calling subroutine MODVIS. A brief description of each subroutine is given next.

Subroutine GRIDG

Before calling this subroutine, the coordinates of all grid nodes, defined in reference to a fixed Cartesian coordinate frame are read. Figure 3 shows the position of cell and grid nodes. The west-to-east, south-to-north and bottom-to-top directions correspond to the ascending indexing order of i, j and k, respectively, forming a right-handed coordinate system.

This subroutine is called only once to calculate coordinates of grid nodes (intersection of grid lines) and geometrical properties of the grid (cell volumes, interpolation factors, normal distances of near-boundary cell nodes from boundary). All variables including grid node coordinates are converted to one-dimensional arrays. The position of any node in one-dimensional array is therefore defined as;

$$IJK = (I-1)*NJ + (K-1)*NI*NJ + J$$

where NI, NJ and NK are the maximum number of grid nodes in the i, j and k directions respectively.

The actual number of grid nodes is one row and one column less than for all cell nodes. For I = NI, J = NJ and K = NK fictitious grid nodes are introduced which have the same coordinates as actual nodes on NI-1 in I-direction, NJ-1 in the J-direction and NK-1 in K-direction.

The subroutine then calculates interpolation factors which are associated with cell nodes and are used in the main program to calculate values of dependent variables at locations other than cell nodes (cell centers). Cell volumes are calculated next followed by calculations of normal distances of near-boundary nodes from all the six outer boundaries.

Subroutine CALCE (PHI, IPHI)

This subroutine solves the linearized and discretized transport equations for the turbulent energy k and the energy dissipation rate ϵ . The two dummy parameters in the calling statement, PHI and IPHI, represent arrays containing dependent variables for which the equation is to be solved. The subroutine sets up the convective and diffusive coefficients over the entire field, then it calculates the source terms for either k or ϵ transport equations. A call is made to MODKE or MODED in order to modify the sources for k and ϵ equations respectively.

The discretized equations have the form

$$A_p \Phi_p = \sum_{i=EWNSTB} A_i \Phi_i + S_\Phi$$

where the coefficients A_i ($i=E,W,N,S,T,B$) contain both the convective and diffusive fluxes. these equations are assembled and solved by calling subroutine SOLSIP which is based on Stone's Strongly Implicit Solver [4].

Subroutine SOLSIP

This subroutine solves the system of linear algebraic equations for k and ϵ using Stone's Implicit Procedure [4]. The array RES (IJK) is used to store the residuals. The sum of absolute residuals "RES1" calculated in the first pass through this part of the routine is used as a measure of convergence of the solution process as a whole and this value is stored in RESOR (IPHI). This variable RESOR (IPHI) is passed to the main solver and if desired can be normalized and compared with the maximum error allowed there. If necessary inner iterations counter L and the sum of absolute residuals RES1 are printed out to monitor the rate of convergence of k and ϵ solution. If the ratio RSM is greater than the maximum allowed for the variable in question, SOR (IPHI), and the number of inner iterations is smaller than a prescribed maximum, NSWP (IPHI), then the routine repeats the sequence of calculating the residuals, increment vectors and updating the dependent variable.

Subroutine MODVIS

This section calculates the effective viscosity and is called after calculating k and ϵ . At locations where ϵ is close to zero (i.e., $\leq 10^{-30}$) viscosity is set to zero. A provision is made for under relaxing changes in effective viscosity which may help to stabilize oscillations and improve convergence rate.

Subroutines MODK and MODED

These subroutines are called from subroutine CALCE and they set the boundary conditions for k and ϵ . For the kinetic energy equation for example, the bottom boundary is checked first for one of the options below;

- (1) An inflow boundary $BCFB(IJ) = 1$ ($IJ = (I-1)*NJ+J$), where the source term is set to accept the inlet values at the x-y plane (bottom boundary $K=1$).
- (2) Outflow boundary $BCFB(IJ) = 2$, where zero gradient in the z-direction is employed.
- (3) Symmetry boundary, $BCFB(IJ) = 3$, where gradients normal to symmetry x-y plane are zero.
- (4) Wall boundary, $BCFB(IJ) = 4$, where the turbulent kinetic energy production (per unit volume) term $GENTB(I)$ calculated from subroutine WALLFN in program MODIFY is added to the rest of the source term $SU(IJK)$.

Boundary conditions for the ϵ -equation are similar to those of k except at the wall where they are set to appropriate values for each near wall treatment.

Program MODIFY

This program is compiled separately and is called from the u, v and w momentum solver. It basically updates the flux source term of the discretized momentum equation due to wall shear stresses. If the u-momentum equation for example is discretized in the form

$$A_p^* u_p = \sum_{i=EWNSTB} A_i u_i + S_u^*$$

where P, E, W, N, S, T, B are cell nodes, and A_p^* and A_i 's contain convective and diffusive coefficients. S_u^* is the source term containing pressure gradients and cross-derivative diffusion terms and convective terms for second-order upwinding scheme. This source term is usually linearized as $S_u^* = S_u - B_p u_p$. The term B_p is usually moved to the left hand side of the equation and modifies the diagonal coefficient $A_p = A_p^* + B_p$, and the equation can be written as

$$A_p u_p = \sum_{i=EWNSTB} A_i u_i + S_u$$

Then S_u and B_p are passed to subroutine MODIFY where they are modified if a wall is present (e.g., BCFB(IJ) = 4 for bottom boundary).

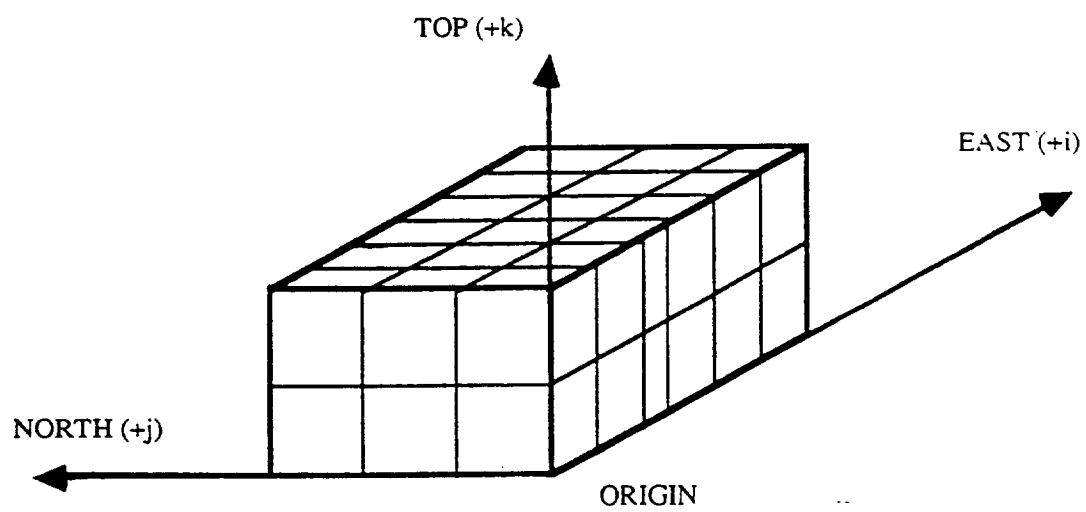


Figure 3. Cell volume and coordinate system

Oct 12 1996 16:42	sskmod_3d	Page 3
143	* VOL1 (DXAF, DYAF, DZAF, DXAE, DYAE, DZAE, DXAH, DYAH, DZAH) +	
144	* VOL1 (DXAE, DYAE, DZAE, DXAG, DYAG, DZAG, DXAH, DYAH, DZAH)	
145	* VOL (IJK) = VOLUM*SIKR	
146	400 CONTINUE	
147	C-----CALCULATIN OF INTERPOLATION FACTORS	
148	C	
149	DO 450 IJK=1,NIJK	
150	FX(IJK)=0.	
151	FY(IJK)=0.	
152	FZ(IJK)=0.	
153	450 FZ(IJK)=0.	
154	C-----KSI - DIRECTION	
155	DO 500 K=2,NKM	
156	LKK=LK(K)	
157	DO 500 J=2,NJM	
158	DO 500 I=2,NIM	
159	IJK=LKK+LI(I)+J	
160	IJKM=IJK-NIJ	
161	IPJK=IJK-NJ	
162	IMJK=IJK-NJ	
163	IPJKM=IJKM-NJ	
164	IMJKM=IJKM-NJ	
165	DKSP=QTR*(X(IJK)+X(IJK-1)+X(IJKM)+X(IJKM-1)-	
166	X(IMJK)-X(IMJK-1)-X(IMJKM)-X(IMJKM-1))	
167	DKKSE=QTR*(X(IPJK)+X(IPJK-1)+X(IPJKM)+X(IPJKM-1)-	
168	X(IJK)-X(IJK-1)-X(IJKM)-X(IJKM-1))	
169	DYKSP=QTR*(Y(IJK)+Y(IJK-1)+Y(IJKM)+Y(IJKM-1)-	
170	Y(IMJK)-Y(IMJK-1)-Y(IMJKM)-Y(IMJKM-1))	
171	DYKSE=QTR*(Y(IPJK)+Y(IPJK-1)+Y(IPJKM)+Y(IPJKM-1)-	
172	Y(IJK)-Y(IJK-1)-Y(IJKM)-Y(IJKM-1))	
173	DZKSP=QTR*(Z(IJK)+Z(IJK-1)+Z(IJKM)+Z(IJKM-1)-	
174	Z(IMJK)-Z(IMJK-1)-Z(IMJKM)-Z(IMJKM-1))	
175	DZKSE=QTR*(Z(IPJK)+Z(IPJK-1)+Z(IPJKM)+Z(IPJKM-1)-	
176	Z(IJK)-Z(IJK-1)-Z(IJKM)-Z(IJKM-1))	
177	DKSP=QTR*(DKKSP**2+DYKSP**2+DZKSP**2)	
178	DKSE=QTR*(DKKSE**2+DYKSE**2+DZKSE**2)	
179	FX(IJK)=DKSP/(DKSP+DKSE)	
180	500 CONTINUE	
181	DO 520 I=1,NIM	
182	LI=LI(I)	
183	DO 520 J=2,NJM	
184	IJK=LI+J	
185	FX(IJK)=FX(IJK-NIJ)	
186	IJK=LK(NK)+IJK	
187	DO 530 K=1,NK	
188	DO 530 K=1,NK	
189	LKK=LK(K)	
190	DO 530 I=1,NIM	
191	IJK=LKK+LI(I)+1	
192	FX(IJK)=FX(IJK+1)	
193	IJK=IJK+NJM	
194	530 FX(IJK)=FX(IJK-1)	
195	C-----ETA - DIRECTION	
196	DO 600 K=2,NKM	
197	LKK=LK(K)	
198	DO 600 I=2,NIM	
199	LI=LI(I)	
200	DO 600 J=2,NJM	
201	IJK=LKK+LI(I)+J	
202	IMJK=IJK-NJ	
203	IJKM=IJK-NIJ	
204	IMJKM=IJKM-NJ	
205	DXETP=QTR*(X(IJK)+X(IMJK)+X(IJKM)+X(IMJKM-1)-	
206	X(IJK-1)-X(IMJK-1)-X(IJKM-1)-X(IMJKM-1))	
207	DXETN=QTR*(X(IJK+1)+X(IMJK+1)+X(IJKM+1)+X(IMJKM+1)-	
208	X(IJK)-X(IMJK)-X(IJKM)-X(IMJKM))	
209	DYETP=QTR*(Y(IJK)+Y(IMJK)+Y(IJKM)+Y(IMJKM-1)-	
210	Y(IJK-1)-Y(IMJK-1)-Y(IJKM-1)-Y(IMJKM-1))	
211	DYETN=QTR*(Y(IJK+1)+Y(IMJK+1)+Y(IJKM+1)+Y(IMJKM+1)-	
212	Y(IJK)-Y(IMJK)-Y(IJKM)-Y(IMJKM))	
213	DZETP=QTR*(Z(IJK)+Z(IMJK)+Z(IJKM)+Z(IMJKM-1)-	

Oct 12 1996 16:42	sskmod_3d	Page 4
214	* Z(IJK-1)-Z(IMJK-1)-Z(IJKM-1)-Z(IMJKM-1))	
215	* Z(IJK)+Z(IMJK)+Z(IJKM+1)+Z(IMJKM+1)-Z(IMJKM+1)-	
216	Z(IJK)-Z(IMJK)-Z(IJKM)-Z(IMJKM))	
217	DETP=QTR*(DXETP**2+DYETP**2+DZETP**2)	
218	DETN=QTR*(DXETN**2+DYETN**2+DZETN**2)	
219	FY(IJK)=DETP/(DETP+DETN)	
220	600 CONTINUE	
221	DO 620 I=2,NIM	
222	LI=LI(I)	
223	DO 620 J=1,NJM	
224	IJK=LI+J	
225	FY(IJK)=FY(IJK-NIJ)	
226	IJK=LK(NK)+IJK	
227	620 FY(IJK)=FY(IJK-NIJ)	
228	DO 630 K=1,NK	
229	LKK=LK(K)	
230	DO 630 J=1,NJM	
231	IJK=LKK+J	
232	FY(IJK)=FY(IJK+NJ)	
233	IJK=LI(NI)+IJK	
234	630 FY(IJK)=FY(IJK-NJ)	
235	C-----ZETA - DIRECTION	
236	DO 700 I=2,NIM	
237	LI=LI(I)	
238	DO 700 J=2,NJM	
239	DO 700 K=2,NKM	
240	IJK=LK(K)+LI+J	
241	IJKM=IJK-NIJ	
242	IJKP=IJK-NJ	
243	IMJK=IJK-NJ	
244	IMJKM=IJKM-NJ	
245	IMJKP=IJKP-NJ	
246	DXZDP=QTR*(X(IJK)+X(IMJK)+X(IJKM)+X(IMJK-1)-	
247	X(IJKM)-X(IMJKM)-X(IJKM-1)-X(IMJKM-1))	
248	DXZDT=QTR*(X(IJKP)+X(IJKP-1)+X(IMJKP)+X(IMJKP-1)-	
249	X(IJK)-X(IJK-1)-X(IMJK)-X(IMJK-1))	
250	DYZDP=QTR*(Y(IJK)+Y(IMJK)+Y(IJKM)+Y(IJK-1)-Y(IMJK-1)-	
251	Y(IJKM)-Y(IMJKM)-Y(IJKM-1)-Y(IMJKM-1))	
252	DYZDT=QTR*(Y(IJKP)+Y(IJKP-1)+Y(IMJKP)+Y(IMJKP-1)-	
253	Y(IJK)-Y(IJK-1)-Y(IMJK)-Y(IMJK-1))	
254	DZDZP=QTR*(Z(IJK)+Z(IMJK)+Z(IJKM)+Z(IJK-1)+Z(IMJK-1)-	
255	Z(IJKM)-Z(IMJKM)-Z(IJKM-1)-Z(IMJKM-1))	
256	DZDZDT=QTR*(Z(IJKP)+Z(IJKP-1)+Z(IMJKP)+Z(IMJKP-1)-	
257	Z(IJK)-Z(IJK-1)-Z(IMJK)-Z(IMJK-1))	
258	DZDP=QTR*(DXZDP**2+DYZDP**2+DZDZP**2)	
259	DZDT=QTR*(DXZDT**2+DYZDT**2+DZDZT**2)	
260	FZ(IJK)=DZDP/(DZDP+DZDT)	
261	700 CONTINUE	
262	DO 720 K=1,NKM	
263	LKK=LK(K)	
264	DO 720 J=2,NJM	
265	IJK=LKK+J	
266	FZ(IJK)=FZ(IJK+NJ)	
267	IJK=LI(NI)+IJK	
268	720 FZ(IJK)=FZ(IJK-NJ)	
269	DO 730 K=1,NKM	
270	LKK=LK(K)	
271	DO 730 I=1,NI	
272	IJK=LKK+LI(I)+1	
273	FZ(IJK)=FZ(IJK+1)	
274	IJK=IJK+NJM	
275	730 FZ(IJK)=FZ(IJK-1)	
276	C	
277	C-----NORMAL DISTANCES FROM DOMAIN BOUNDARIES	
278	C	
279	DO 740 IJ=1,NIJ	
280	DNB(IJ)=0.	
281	DNT(IJ)=0.	
282	NIK=NI*NK	
283	NJK=NJ*NK	
284	DO 741 IK=1,NIK	

Oct 12 1996 16:42		sskmod_3d	Page 5
285		DNS(IK)=0.	
286	741	DNN(IK)=0.	
287		DO 742 JK=1,NJK	
288		DNN(JK)=0.	
289	742	DNE(JK)=0.	
290		DO 750 I=2,NIM	
291		LII=LI(I)	
292		DO 760 J=2,NJM	
293	C-----	BOTTOM BOUNDARY	
294		IQ=LII+J	
295		IJK=LK(2)+IJ	
296		IJKM=IJK-NJ	
297		IMJK=IJK-NJ	
298		IMJRM=IJKM-NJ	
299		DX1=X(IJRM)-X(IMJRM)	
300		DY1=Y(IJRM)-Y(IMJRM)	
301		DZ1=Z(IJRM)-Z(IMJRM)	
302		DX2=X(IJRM)-X(IMJRM)	
303		DY2=Y(IJRM)-Y(IMJRM)	
304		DZ2=Z(IJRM)-Z(IMJRM)	
305		XB=QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1))	
306		YB=QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1))	
307		ZB=QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1))	
308		XP=HAF*(XB+QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1)))	
309		YP=HAF*(YB+QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1)))	
310		ZP=HAF*(ZB+QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1)))	
311		DX3=XP-YB	
312		DY3=YB-YB	
313		DZ3=ZP-ZB	
314		DNB(IJ)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)	
315	C-----	TOP BOUNDARY	
316		IJK=LK(NKM)+IJ	
317		IJKM=IJK-NIJ	
318		IMJK=IJK-NJ	
319		IMJRM=IJKM-NJ	
320		DX1=X(IJRM)-X(IMJRM)	
321		DY1=Y(IJRM)-Y(IMJRM)	
322		DZ1=Z(IJRM)-Z(IMJRM)	
323		DX2=X(IJRM)-X(IMJRM)	
324		DY2=Y(IJRM)-Y(IMJRM)	
325		DZ2=Z(IJRM)-Z(IMJRM)	
326		XB=QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1))	
327		YB=QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1))	
328		ZB=QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1))	
329		XP=HAF*(XB+QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1)))	
330		YP=HAF*(YB+QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1)))	
331		ZP=HAF*(ZB+QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1)))	
332		DX3=XP-YB	
333		DY3=YB-YB	
334		DZ3=ZP-ZB	
335		DNT(IJ)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)	
336	760	CONTINUE	
337	C-----	SOUTH BOUNDARY	
338		DO 770 K=2,NKM	
339		IK=(I-1)*NK+K	
340		IJK=LII+LK(K)+2	
341		IJKM=IJK-NIJ	
342		IMJK=IJK-NJ	
343		IMJRM=IJKM-NJ	
344		DX1=X(IJRM)-X(IMJRM)	
345		DY1=Y(IJRM)-Y(IMJRM)	
346		DZ1=Z(IJRM)-Z(IMJRM)	
347		DX2=X(IJRM)-X(IMJRM)	
348		DY2=Y(IJRM)-Y(IMJRM)	
349		DZ2=Z(IJRM)-Z(IMJRM)	
350		XB=QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1))	
351		YB=QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1))	
352		ZB=QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1))	
353		XP=HAF*(XB+QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1)))	
354		YP=HAF*(YB+QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1)))	
355		ZP=HAF*(ZB+QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1)))	

Oct 12 1996 16:42		sskmod_3d	Page 6
356		DX3=XP-YB	
357		DY3=YB-YB	
358		DZ3=ZP-ZB	
359		DNS(IK)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)	
360	C-----	NORTH BOUNDARY	
361		IJK=LK(K)+LII+NJM	
362		IJKM=IJK-NIJ	
363		IMJK=IJK-NJ	
364		IMJRM=IJKM-NJ	
365		DX1=X(IJRM)-X(IMJRM)	
366		DY1=Y(IJRM)-Y(IMJRM)	
367		DZ1=Z(IJRM)-Z(IMJRM)	
368		DX2=X(IJRM)-X(IMJRM)	
369		DY2=Y(IJRM)-Y(IMJRM)	
370		DZ2=Z(IJRM)-Z(IMJRM)	
371		XB=QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1))	
372		YB=QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1))	
373		ZB=QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1))	
374		XP=HAF*(XB+QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1)))	
375		YP=HAF*(YB+QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1)))	
376		ZP=HAF*(ZB+QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1)))	
377		DX3=XP-YB	
378		DY3=YB-YB	
379		DZ3=ZP-ZB	
380		DNN(IK)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)	
381	770	CONTINUE	
382	750	CONTINUE	
383	C-----	WEST BOUNDARY	
384		DO 780 K=2,NKM	
385		LKF=LK(K)	
386		DO 780 J=2,NJM	
387		JK=(K-1)*NJ+J	
388		IJK=LKK+LI(2)+J	
389		IJKM=IJK-NIJ	
390		IMJK=IJK-NJ	
391		IMJRM=IJKM-NJ	
392		DX1=X(IJRM)-X(IMJRM)	
393		DY1=Y(IJRM)-Y(IMJRM)	
394		DZ1=Z(IJRM)-Z(IMJRM)	
395		DX2=X(IJRM)-X(IMJRM)	
396		DY2=Y(IJRM)-Y(IMJRM)	
397		DZ2=Z(IJRM)-Z(IMJRM)	
398		XB=QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1))	
399		YB=QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1))	
400		ZB=QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1))	
401		XP=HAF*(XB+QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1)))	
402		YP=HAF*(YB+QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1)))	
403		ZP=HAF*(ZB+QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1)))	
404		DX3=XP-YB	
405		DY3=YB-YB	
406		DZ3=ZP-ZB	
407		DNN(JK)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)	
408	C-----	EAST BOUNDARY	
409		IJK=LKK+LI(NIM)+J	
410		IJKM=IJK-NIJ	
411		IMJK=IJK-NJ	
412		IMJRM=IJKM-NJ	
413		DX1=X(IJRM)-X(IMJRM)	
414		DY1=Y(IJRM)-Y(IMJRM)	
415		DZ1=Z(IJRM)-Z(IMJRM)	
416		DX2=X(IJRM)-X(IMJRM)	
417		DY2=Y(IJRM)-Y(IMJRM)	
418		DZ2=Z(IJRM)-Z(IMJRM)	
419		XB=QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1))	
420		YB=QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1))	
421		ZB=QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1))	
422		XP=HAF*(XB+QTR*(X(IJRM)+X(IJRM-1)+X(IMJRM)+X(IMJRM-1)))	
423		YP=HAF*(YB+QTR*(Y(IJRM)+Y(IJRM-1)+Y(IMJRM)+Y(IMJRM-1)))	
424		ZP=HAF*(ZB+QTR*(Z(IJRM)+Z(IJRM-1)+Z(IMJRM)+Z(IMJRM-1)))	
425		DX3=XP-YB	
426		DY3=YB-YB	

Oct 12 1996 16:42 sskemod_3d Page 7

```

427 DZ3=ZP-ZB
428 DNE(JK)=DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)
429 780 CONTINUE
430 C
431 C CALCULATE THE NORMAL DISTANCES FROM BOUNDARIES
432 C
433 FUNCTION DELTA(DX1,DX2,DX3,DY1,DY2,DY3,DZ1,DZ2,DZ3)
434 XAN=DY1*DZ2-DY2*DZ1
435 YAN=DX2*DZ1-DX1*DZ2
436 ZAN=DX1*DY2-DX2*DY1
437 ARNR=1./SQRT(XAN**2+YAN**2+ZAN**2)
438 DELN=(DX3*XAN+DY3*YAN+DZ3*ZAN)*ARNR
439 RETURN
440 END
441 C
442 C-----
443 SUBROUTINE CALCE (PHI,IPHI)
444 INCLUDE 'kernod.h'
445 C
446 DIMENSION PHIB(JY),PHI(JXYZ)
447 INTEGER BCFW,BCFE,BCFS,BCFN,BODB,BCFT
448 C-----INITIALISE TEMPORARILY STORED VARIABLES
449 C
450 C
451 G=GKE
452 IF (IPHI.EQ.1) THEN
453   URFPFI=1./URFK
454   PRVINVP=1./PRTK
455   ELSE
456   URFPFI=1./URFE
457   PRVINVP=1./PRTE
458   ENDIF
459 C
460 DO 5 J=1,NJ
461 5 PHIB(J)=PHI(J)
462 C-----
463 BOTTOM BOUNDARY CELL FACES
464 LII=LI(1)
465 PHIS=PHI(LII+1)
466 DO 10 J=2,NJM
467   IJ=LII+J
468   IJK=IJ
469   IJKP=IJK-NIJ
470   IMJK=IJK-NJ
471   IMJMK=IJK-1
472   IMJMK=IMJK-1
473   VOLT=HAF*(VOL(IJK)+VOL(IJMK)+X(IJMK)-X(IMJMK))
474   DXKS=HAF*(X(IJK)-X(IMJK)+X(IJMK)+X(IMJMK))
475   DXET=HAF*(X(IJK)-X(IJMK)+X(IJMK)-X(IMJMK))
476   DXZD=HAF*(X(IJKP)+X(IJMK)-X(IJMK)-X(IMJMK))
477   &
478   DYKS=HAF*(Y(IJK)-Y(IMJK)+Y(IJMK)+Y(IMJMK))
479   DYET=HAF*(Y(IJK)-Y(IJMK)+Y(IJMK)-Y(IMJMK))
480   DYZD=HAF*(Y(IJKP)+Y(IJMK)-Y(IJMK)-Y(IMJMK))
481   &
482   DZKS=HAF*(Z(IJK)-Z(IMJK)+Z(IJMK)+Z(IMJMK))
483   DZET=HAF*(Z(IJK)-Z(IJMK)+Z(IJMK)-Z(IMJMK))
484   DZD=HAF*(Z(IJKP)+Z(IJMK)-Z(IJMK)-Z(IMJMK))
485   &
486   ZB(IJ)-ZB(IJ-1)-ZB(IJ-NJ)-ZB(IJ-NJ-1)
487   B11=DYET*DZD-DYZD*DZET
488   B12=DXZD*DZET-DXET*DZD
489   B13=DXET*DYD-DYET*DXD
490   B21=DZKS*DYD-DZD*DYKS
491   B22=DXKS*DZD-DXD*DXKS
492   B23=DXZD*DYKS-DYKS*DXZD
493   B31=DYKS*DZET-DYET*DZKS
494   B32=DZKS*DYET-DZET*DYKS
495   B33=DXKS*DYET-DXET*DYKS
496   &
497   ART=B31**2+B32**2+B33**2

```

sskemod_3d

Oct 12 1996 16:42 sskemod_3d Page 8

```

498 GAWT=VIS(IJK)*PRTINVP/VOLT
499 DB(IJ)=GAWT*ART
500 FZBB(IJ)=1.0
501 PHIE=PHI(IJK)*(1.-FX(IJK))+PHI(IJK+NJ)*FX(IJK)
502 PHIN=PHI(IJK)*(1.-FY(IJK))+PHI(IJK+1)*FY(IJK)
503 SUB(IJ)=GAWT*((B11*B31+B12*B32+B13*B33)*(PHIE-PHIN)+
504   (B21*B31+B22*B32+B23*B33)*(PHIN-PHIS))
505 PHIS=PHIN
506 PHINB(J)=PHIE
507 10 CONTINUE
508 C-----WEST BOUNDARY CELL FACES
509 DO 100 K=2,NKM
510   LKK=LK(K)
511   KKJ=(K-1)*NJ
512   KKI=(K-1)*NI
513   PHIS=PHI(LKK+1)
514   DO 20 J=2,NJM
515     JK=KKJ+J
516     IJK=LKK+J
517     IMJK=IJK-1
518     IMJMK=IJK-NIJ
519     IMJMK=IMJK-1
520     VOL=HAF*(VOL(IJK)+VOL(IJMK)+X(IJMK+NJ)+X(IMJMK+NJ)-
521       &
522       XZW(JK)-XZW(JK-1)-XZW(JK-NJ)-XZW(JK-NJ-1))
523     DXET=HAF*(X(IJK)-X(IJMK)+X(IJMK)+X(IMJMK))
524     DXZD=HAF*(X(IJK)-X(IJMK)+X(IJMK)-X(IMJMK))
525     DYKS=HAF*(Y(IJK+NJ)+Y(IMJMK+NJ)+Y(IJMK+NJ)+Y(IMJMK+NJ)-
526       &
527       YZW(JK)-YZW(JK-1)-YZW(JK-NJ)-YZW(JK-NJ-1))
528     DYET=HAF*(Y(IJK)-Y(IMJK)+Y(IJMK)+Y(IMJMK))
529     DZKS=HAF*(Z(IJK+NJ)+Z(IMJMK+NJ)+Z(IJMK+NJ)+Z(IMJMK+NJ)-
530       &
531       ZZW(JK)-ZZW(JK-1)-ZZW(JK-NJ)-ZZW(JK-NJ-1))
532     DZET=HAF*(Z(IJK)-Z(IMJK)+Z(IJMK)+Z(IMJMK))
533     DZD=HAF*(Z(IJK)-Z(IJMK)+Z(IJMK)-Z(IMJMK))
534     &
535     INLINE
536     B11=DYET*DZD-DYZD*DZET
537     B12=DXZD*DZET-DXET*DZD
538     B13=DXET*DYD-DYET*DXD
539     B21=DZKS*DYD-DZD*DYKS
540     B22=DXKS*DZD-DXD*DXKS
541     B23=DXZD*DYKS-DYKS*DXZD
542     B31=DYKS*DZET-DYET*DZKS
543     B32=DZKS*DYET-DZET*DYKS
544     B33=DXKS*DYET-DXET*DYKS
545     &
546     INLINE
547     ARE=B11**2+B12**2+B13**2
548     GAME=VIS(IJK)*PRTINVP/VOLT
549     DW(J)=GAME*ARE
550     FXW(J)=1.0
551     PHIN=PHI(IJK)*(1.-FY(IJK))+PHI(IJK+1)*FY(IJK)
552     PHIB=PHI(IJK)*(1.-FZ(IJK))+PHI(IJK+NJ)*FZ(IJK)
553     PHINB(J)=PHI(IJMK)+PHI(IJMK+NJ)*FZ(IJMK)
554     SUM(J)=GAME*((B21*B11+B22*B12+B23*B13)*(PHIN-PHIS)+
555       (B31*B11+B32*B12+B33*B13)*(PHIB-PHIB))
556     PHIS=PHIN
557 20 CONTINUE
558 C-----SOUTH BOUNDARY CELL FACES
559 DO 100 I=2,NIM
560   IK=(I-1)*NK+K
561   LII=LI(1)
562   IJK=LKK+LII+1
563   IMJK=IJK-NIJ
564   IMJMK=IJK-NJ
565   VOLN=HAF*(VOL(IJK)+VOL(IJMK)+X(IJMK)+X(IMJMK)-X(IMJMK+NJ)-
566     &
567     DXKS=HAF*(X(IJK)-X(IMJMK+NJ)+X(IJMK+NJ)+X(IMJMK+NJ)-
568     &
569     XZS(IK)-XZS(IK-1)-XZS(IK-NK)-XZS(IK-NK-1))
570     DXET=HAF*(X(IJK)-X(IJMK)+X(IJMK)+X(IMJMK))
571     DXZD=HAF*(X(IJK)-X(IJMK)+X(IJMK)-X(IMJMK))
572     DYKS=HAF*(Y(IJK)+Y(IMJMK)+Y(IJMK)+Y(IMJMK))
573     DYET=HAF*(Y(IJK)-Y(IMJMK)+Y(IJMK)-Y(IMJMK))
574     DZKS=HAF*(Z(IJK)+Z(IMJMK)+Z(IJMK)+Z(IMJMK))
575     DZET=HAF*(Z(IJK)-Z(IMJMK)+Z(IJMK)-Z(IMJMK))
576     &
577     INLINE
578     ART=B31**2+B32**2+B33**2

```

Oct 12 1996 16:42 **sskmod_3d** Page 9

```

569 DYET=AHT*(Y(IJK+1)+Y(IMJK+1)+Y(IJMK+1)+Y(IMJMK+1))-
570 & ZS(IK)-ZS(IK-1)-ZS(IK-NK)-ZS(IK-NK-1))
571 DYZD=HAF*(Y(IJK)-Z(IMJK)+Y(IMJK)-Y(IMJMK))
572 DZKS=HAF*(Z(IJK)-Z(IMJK)+Z(IJMK)-Z(IMJMK))
573 DZET=AHT*(Z(IJK+1)+Z(IMJK+1)+Z(IJMK+1)+Z(IMJMK+1))-
574 & ZS(IK)-ZS(IK-1)-ZS(IK-NK)-ZS(IK-NK-1))
575 DZZD=HAF*(Z(IJK)-Z(IJMK)+Z(IMJK)-Z(IMJMK))
576 C.....,INLINE
577 B11=DYET*DZZD-DYZD*DZET
578 B12=DXZD*DZET-DXET*DZZD
579 B13=DXET*DYZD-DYET*DXZD
580 B21=DXKS*DYZD-DZZD*DYKS
581 B22=DXKS*DZZD-DXZD*DXKS
582 B23=DXZD*DYKS-DXKS*DYZD
583 B31=DYKS*DZET-DYET*DXKS
584 B32=DXKS*DZET-DZET*DXKS
585 B33=DXKS*DYET-DXET*DYKS
586 C.....,INLINE
587 ARN=B21**2+B22**2+B23**2
588 GARN=VIS(IJK)*PRTINVP/VOLN
589 DN=GARN*ARN
590 FYSS=1.0
591 PHIE=PHI(IJK)*(1.-FX(IJK))+PHI(IJK+NJ)*FX(IJK)
592 PHIW=PHI(IMJK)*(1.-FX(IMJK))+PHI(IJK)*FX(IMJK)
593 PHIT=PHI(IJK)*(1.-FZ(IJK))+PHI(IJK+NIJ)*FZ(IJK)
594 PHUB=PHI(IJMK)*(1.-FZ(IJMK))+PHI(IJK)*FZ(IJMK)
595 SUN=GARN*(B11*B21+B12*B22+B13*B23)*(PHIE-PHIW)+
596 * (B31*B21+B32*B22+B33*B23)*(PHIT-PHIB))
597 C
598 C-----THE MAIN LOOP
599 C
600 DO 100 J=2,NJM
601 IG=LII+J
602 IJK=LAK+IJ
603 IJPK=IJK+NIJ
604 IJMK=IJK+NIJ
605 IPJK=IJK+NJ
606 IMJK=IJK-NJ
607 IJPK=IJK+1
608 IJMK=IJK-1
609 FXE=FX(IJK)
610 FXN=1.-FXE
611 FYN=FY(IJK)
612 FYS=1.-FYN
613 FZT=FZ(IJK)
614 FZB=1.-FZT
615 C
616 VOLE=HAF*(VOL(IJK)+VOL(IPJK))
617 VOLN=HAF*(VOL(IJK)+VOL(IJPK))
618 VOLT=HAF*(VOL(IJK)+VOL(IJPK))
619 GAME=(VIS(IJK)*FXW-VIS(IPJK)*FYE)*PRTINVP/VOLE
620 GAMN=(VIS(IJK)*FYS-VIS(IJPK)*FYN)*PRTINVP/VOLN
621 GAMT=(VIS(IJK)*FZB-VIS(IJPK)*FZT)*PRTINVP/VOLT
622 C
623 C-----EAST CELL FACE
624 C
625 DXKS=AHT*(X(IPJK)+X(IJPK-1)+X(IJMK+NJ)+X(IJMK+NJM)-
626 * X(IMJK)-X(IMJK-1)-X(IJMK-NJ)-X(IJMK-NJ-1))
627 DYET=HAF*(X(IJK)-X(IJMK)+X(IJMK)-X(IJMK-1))
628 DXZD=HAF*(X(IJK)-X(IJMK)-X(IJMK)-X(IJMK-1))
629 DYKS=AHT*(X(IPJK)+X(IJPK-1)+X(IJMK+NJ)+X(IJMK+NJM)-
630 * Y(IMJK)-Y(IMJK-1)-Y(IJMK-NJ)-Y(IJMK-NJ-1))
631 DYET=HAF*(Y(IJK)-Y(IJMK)+Y(IJMK)-Y(IJMK-1))
632 DYZD=HAF*(Y(IJPK)+Y(IJPK-1)+Y(IJMK+NJ)+Y(IJMK+NJM)-
633 * Z(IMJK)-Z(IMJK-1)-Z(IJMK-NJ)-Z(IJMK-NJ-1))
634 DZKS=AHT*(Z(IPJK)+Z(IJPK-1)+Z(IJMK+NJ)+Z(IJMK+NJM)-
635 * Z(IMJK)-Z(IMJK-1)-Z(IJMK-NJ)-Z(IJMK-NJ-1))
636 DZZD=HAF*(Z(IJK)-Z(IJMK)+Z(IJMK)-Z(IJMK-1))
637 C.....,INLINE
638 B11=DYET*DZZD-DYZD*DZET
639 B12=DXZD*DZET-DXET*DZZD

```

Oct 12 1996 16:42 **sskmod_3d** Page 10

```

640 B13=DXET*DY2D-DYET*DX2D
641 B21=DZKS*DY2D-DZ2D*DYKS
642 B22=DYKS*DZ2D-DX2D*DZKS
643 B23=DX2D*DYKS-DXKS*DY2D
644 B31=DYKS*DZET-DYET*DZKS
645 B32=DZKS*DXET-DZET*DXKS
646 B33=DXKS*DYET-DXET*DYKS
647 C.....,INLNE
648 ARE=B11*2+B12*2+B13*2
649 DE=GAME*ARE
650 AE=AMINI(F1(IJK),0.)*FX(IPJK)*G
651 AWE=AMAXI(F1(IMJK),0.)*(1.-FXHW(J))*G
652 AEI=AMINI(F1(IMJK),0.)*FXE*G
653 AEI(JK)=DE-AMINI(F1(IJK),0.)*-AEE
654 AWI=AMAXI(F1(IJK),0.)*(1.-FX(IMJK))*G
655 AW(IJK)=DW(J)+AMAXI(F1(IMJK),0.)*-AWW
656 C
657 PHTE=PHI(IJK)*FXW+PHI(IPJK)*FXE
658 PHTEB=(PHI(IJMK)*FXW+PHI(IJMK+NU)*FXE)*(1.-FZ(IJMK))+PHIE*FZ(IJMK)
659 PHTEB=(PHI(IJMK)*FXW+PHI(IJMK+NU)*FXE)*FZT+PHIE*FZB
660 PHISE=(PHI(IJMK)*FXW+PHI(IPJK-1)*FXE)*(1.-FY(IJMK))+PHIE*FY(IJMK)
661 PHINEB=(PHI(IJPK)*FXW+PHI(IPJK+1)*FXE)*FYN+PHIE*FYS
662 SUE=GAME*(B11*B21+B12*B22+B13*B23)*(PHINE-PHIEB)+
663 * (B11*B31+B12*B32+B13*B33)*(PHITE-PHIEB)
664 C
665 C-----NORTH CELL - FACE
666 C
667 DXKS=HAF*(X(IJK)-X(IMJK)+X(IJMK)-X(IJMK-NJ))
668 DXET=AHF*(X(IJPK)-X(IMJK-1)+X(IJMK-1)+X(IJMK-NJM)-
669 * X(IJMK)+X(IMJK-1)-X(IJMK-1)-X(IJMK-NJ-1))
670 DX2D=HAF*(X(IJJK)-X(IJMK)+X(IJMK)-X(IJMK-NJ))
671 DYKS=HAF*(Y(IJJK)-Y(IMJK)+Y(IJMK)-Y(IJMK-NJ))
672 DYET=AHF*(Y(IJPK)+Y(IMJK+1)+Y(IJMK-1)+Y(IJMK-NJM)-
673 * Y(IJMK)-Y(IMJK-1)-Y(IJMK-1)-Y(IJMK-NJ-1))
674 DY2D=HAF*(Y(IJJK)-Y(IJMK)+Y(IMJK)-Y(IJMK-NJ))
675 DZKS=HAF*(Z(IJJK)-Z(IJMK)+Z(IJMK)-Z(IJMK-NJ))
676 DZET=AHF*(Z(IJPK)+Z(IMJK+1)+Z(IJMK-1)+Z(IJMK-NJM)-
677 * Z(IJMK)-Z(IMJK-1)-Z(IJMK-1)-Z(IJMK-NJ-1))
678 DZ2D=HAF*(Z(IJJK)-Z(IJMK)+Z(IJMK)-Z(IJMK-NJ))
679 C.....,INLNE
680 ARN=B21*2+B22*2+B23*2
681 DS=DN
682 DN=GAMN*ARN
683 ANN=AMINI(F2(IJK),0.)*FY(IJPK)*G
684 ASS=AMAXI(F2(IJMK),0.)*(1.-FYSS)*G
685 ANI=AMINI(F2(IJMK),0.)*FYN*G
686 AN(IJK)=DN-AMINI(F2(IJK),0.)*-ANN
687 ASI=AMAXI(F2(IJK),0.)*(1.-FY(IJMK))*G
688 AS(IJK)=DS+AMAXI(F2(IJMK),0.)*-ASS
689 C.....,INLNE
690 ARN=B21*2+B22*2+B23*2
691 DS=DN
692 DN=GAMN*ARN
693 ANN=AMINI(F2(IJK),0.)*FY(IJPK)*G
694 ASS=AMAXI(F2(IJMK),0.)*(1.-FYSS)*G
695 ANI=AMINI(F2(IJMK),0.)*FYN*G
696 AN(IJK)=DN-AMINI(F2(IJK),0.)*-ANN
697 ASI=AMAXI(F2(IJK),0.)*(1.-FY(IJMK))*G
698 AS(IJK)=DS+AMAXI(F2(IJMK),0.)*-ASS
699 C
700 PHIN=PHI(IJJK)*FYS+PHI(IJPK)*FYN
701 PHITN=(PHI(IJPK)*FYS+PHI(IJPK+1)*FYN)*FZT+PHIN*FZB
702 PHIBN=(PHI(IJMK)*FYS+PHI(IJMK+1)*FYN)*(1.-FZ(IJMK))+PHIN*FZ(IJMK)
703 PHIEB=(PHI(IPJK)*FYS+PHI(IPJK+1)*FYN)*FXE+PHIN*FXW
704 PHIWN=(PHI(IMJK)*FYS+PHI(IMJK+1)*FYN)*(1.-FX(IMJK))+PHIN*FX(IMJK)
705 SWS=SUM
706 SUN=GAMN*(B11*B21+B12*B22+B13*B23)*(PHIEB-PHIBN)+
707 * (B31*B21+B32*B22+B33*B23)*(PHITN-PHIBN)
708 C
709 C-----TOP CELL - FACE
710 C

```

Oct 12 1996 16:42

sskmod_3d

Page 11

```

711 DXKS=HAF*(X(IJK)-X(IMJK)+X(IJMK)-X(IMJK-1))
712 DXET=HAF*(X(IJK)-X(IMJK)+X(IJMK)-X(IMJK-1))
713 DXZD=AMT*(X(IJPK)+X(IJPK-1)+X(IJPK-NJ)+X(IJPK-NJ-1))
714 * X(IJPK)-X(IJMK-1)-X(IJMK-NJ)-X(IJMK-NJ-1))
715 DYKS=HAF*(Y(IJK)-Y(IMJK)+Y(IJMK)-Y(IMJK-1))
716 DYET=HAF*(Y(IJK)-Y(IMJK)+Y(IJMK)-Y(IMJK-1))
717 DYZD=AMT*(Y(IJPK)+Y(IJPK-1)+Y(IJPK-NJ)+Y(IJPK-NJ-1))
718 * Y(IJPK)-Y(IMJK-1)-Y(IMJK-NJ)-Y(IMJK-NJ-1))
719 DZKS=HAF*(Z(IJK)-Z(IMJK)+Z(IJMK)-Z(IMJK-1))
720 DZET=HAF*(Z(IJK)-Z(IMJK)+Z(IJMK)-Z(IMJK-1))
721 DZDZ=AMT*(Z(IJPK)+Z(IJPK-1)+Z(IJPK-NJ)+Z(IJPK-NJ-1))
722 * Z(IJPK)-Z(IMJK-1)-Z(IMJK-NJ)-Z(IMJK-NJ-1))
723 C.....INLINE
724 B11=DYET-DZDZ-DYZD-DZET
725 B12=DXZD-DZET-DZET-DZDZ
726 B13=DXET-DZDZ-DZET-DZDZ
727 B21=DZKS-DZDZ-DZDZ-DZKS
728 B22=DXKS-DZDZ-DZDZ-DZKS
729 B23=DXZD-DZKS-DZKS-DZDZ
730 B31=DZKS-DZET-DZET-DZKS
731 B32=DXKS-DZET-DZET-DZKS
732 B33=DXKS-DZET-DZET-DZKS
733 C.....INLINE
734 ART=B31**2+B32**2+B33**2
735 DT=GAMT*ART
736 ATT=AMIN1(F3(IJK),0)*FZ(IJPK)*G
737 ABB=AMAX1(F3(IJK),0)*(1.-FZB(IJ))**G
738 AT1=-AMIN1(F3(IJK),0)*FZT*G
739 AT(IJK)=DT-AMIN1(F3(IJK),0)*ATT
740 AB1=AMAX1(F3(IJK),0)*(1.-FZ(IJMK))*G
741 AB(IJK)=DB(IJ)+AMAX1(F3(IJK),0)*ABB
742 C
743 PHIT=PHI(IJPK)+FZB*PHI(IJPK-NJ)*FZT
744 PHIT=PHI(IJPK)+FZB*PHI(IJPK-NJ)*FZT*(1.-FX(IMJK))+PHIT*FX(IMJK)
745 PHIT=PHI(IJPK)+FZB*PHI(IJPK-NJ)*FZT*(1.-FX(IMJK))+PHIT*FX(IMJK)
746 PHIT=PHI(IJPK)+FZB*PHI(IJPK-NJ)*FZT*(1.-FX(IMJK))+PHIT*FX(IMJK)
747 PHIT=PHI(IJPK)+FZB*PHI(IJPK-NJ)*FZT*(1.-FX(IMJK))+PHIT*FX(IMJK)
748 * SUT=GAMT*((B11*B31+B12*B32+B13*B33)*(PHIT-PHIT)+
749 (B21*B31+B22*B32+B23*B33)*(PHIT-PHIT))
750 BT(IJK)=SUT-SUT*(J)+SUN-SUT-SUB(IJ)
751 BT(IJK)=AMAX1(0,0,BT(IJK))
752 * BT(IJK)=AEB*PHI(IJPK)+AEB*PHI(IJPK-NJ)+AMW*PHI(IMJK-NJ)+ANN*PHI(IJPK+1)+
753 * AEB*PHI(IMJK-NJ)+ATT*PHI(IJPK-NJ)+AB*PHI(IJPK-NJ)+
754 * AEB*PHI(IJPK)+AM1*PHI(IMJK)+ANI*PHI(IJPK)+AS1*PHI(IMJK)+
755 * AT1*PHI(IJPK)+AB1*PHI(IMJK)
756 SP(IJK)=AEB*ANN+ANN*ASS+ATT*ABB+AE1*AW1+AS1*AT1+AB1
757 SUB(IJ)=SUT
758 SUB(IJ)=SUE
759 FZBB(IJ)=FZ(IJMK)
760 FXW(IJ)=FX(IMJK)
761 FYSS=FY(IMJK)
762 DW(IJ)=DE
763 DB(IJ)=DT
764 C-----GENERATION OF TURB. KINETIC ENERGY
765 DXKS=QTR*(X(IJK)+X(IJMK)+X(IJMK)-X(IJMK-1)-X(IMJK)-X(IMJK-1)-
766 * X(IJMK-NJ)-X(IJMK-NJ-1))
767 DXET=QTR*(X(IJK)+X(IJMK)+X(IMJK)+X(IJMK-NJ)-
768 * X(IMJK)-X(IMJK-1)-X(IJMK-NJ)-X(IJMK-NJ-1))
769 DXZD=QTR*(X(IJK)+X(IJMK)+X(IMJK)+X(IMJK-NJ)-X(IJMK)-
770 * X(IJMK-NJ)-X(IJMK-NJ-1))
771 DYKS=QTR*(Y(IJK)+Y(IMJK)+Y(IJMK)+Y(IJMK-NJ)-Y(IMJK)-Y(IMJK-1)-
772 * Y(IJMK-NJ)-Y(IJMK-NJ-1))
773 DYET=QTR*(Y(IJK)+Y(IMJK)+Y(IMJK)+Y(IMJK-NJ)-Y(IMJK)-Y(IMJK-1)-
774 * Y(IMJK-NJ)-Y(IMJK-NJ-1))
775 DYZD=QTR*(Z(IJK)+Z(IMJK)+Z(IMJK)+Z(IMJK-NJ)-Z(IMJK)-Z(IMJK-1)-
776 * Z(IMJK-NJ)-Z(IMJK-NJ-1))
777 DZKS=QTR*(Z(IJK)+Z(IMJK)+Z(IMJK)+Z(IMJK-NJ)-Z(IMJK)-Z(IMJK-1)-
778 * Z(IMJK-NJ)-Z(IMJK-NJ-1))
779 DZET=QTR*(Z(IJK)+Z(IMJK)+Z(IMJK)+Z(IMJK-NJ)-Z(IMJK)-Z(IMJK-1)-
780 * Z(IMJK-NJ)-Z(IMJK-NJ-1))
781 DZDZ=QTR*(Z(IJK)+Z(IMJK)+Z(IMJK)+Z(IMJK-NJ)-Z(IMJK)-Z(IMJK-1)-

```

sskmod_3d

6

Oct 12 1996 16:42

sskmod_3d

Page 12

```

782 C.....INLINE
783 * Z(IJMK-NJ)-Z(IJMK-1)-Z(IJMK-NJ-1))
784 B11=DYET-DZDZ-DYZD-DZET
785 B12=DXZD-DZET-DZET-DZDZ
786 B13=DXET-DZDZ-DZET-DZDZ
787 B21=DZKS-DZDZ-DZDZ-DZKS
788 B22=DXKS-DZDZ-DZDZ-DZKS
789 B23=DXZD-DZKS-DZKS-DZDZ
790 B31=DZKS-DZET-DZET-DZKS
791 B32=DXKS-DZET-DZET-DZKS
792 B33=DXKS-DZET-DZET-DZKS
793 C.....INLINE
794 C
795 UEW=U(IJPK)+FXE-U(IMJK)*(1.-FX(IMJK))+U(IJK)*(FXW-FX(IMJK))
796 UNS=U(IJPK)+FYN-U(IMJK)*(1.-FY(IMJK))+U(IJK)*(FYS-FY(IMJK))
797 UTB=U(IJPK)+FZT-U(IMJK)*(1.-FZ(IMJK))+U(IJK)*(FZB-FZ(IMJK))
798 VEW=V(IJPK)+FXE-V(IMJK)*(1.-FX(IMJK))+V(IJK)*(FXW-FX(IMJK))
799 VNS=V(IJPK)+FYN-V(IMJK)*(1.-FY(IMJK))+V(IJK)*(FYS-FY(IMJK))
800 VTB=V(IJPK)+FZT-V(IMJK)*(1.-FZ(IMJK))+V(IJK)*(FZB-FZ(IMJK))
801 MEW=W(IJPK)+FXE-W(IMJK)*(1.-FX(IMJK))+W(IJK)*(FXW-FX(IMJK))
802 WNS=W(IJPK)+FYN-W(IMJK)*(1.-FY(IMJK))+W(IJK)*(FYS-FY(IMJK))
803 WTB=W(IJPK)+FZT-W(IMJK)*(1.-FZ(IMJK))+W(IJK)*(FZB-FZ(IMJK))
804 C
805 GEN(IJK)=(2.*((B11*UEW+B21*UNS+B31*UTB)**2+
806 * (B12*VEW+B22*VNS+B32*VTB)**2+(B13*MEW+B23*WNS+B33*WTB)**2+
807 * (B12*UEW+B22*UNS+B32*UTB+B11*VEW+B21*WNS+B31*WTB)**2+
808 * (B13*VEW+B23*VNS+B33*VTB+B12*MEW+B22*WNS+B32*WTB)**2)
809 * / (VOL(IJK)**2)
810 C
811 IF (IPHI.EQ.2) GO TO 102
812 C
813 C-----TURBULENCE KINETIC ENERGY - SOURCE TERMS
814 C
815 SU(IJK)=BT(IJK)+GEN(IJK)*VOL(IJK)*VOL(IJK)*(VIS(IJK)-VISCOS)
816 SP(IJK)=SP(IJK)+VOL(IJK)*CMO*DENSIT*DENSIT*TE(IJK)/
817 * (VIS(IJK)-VISCOS+SMALL)
818 C
819 GO TO 100
820 C
821 C-----DISSIPATION OF TURB. KIN. ENERGY SOURCE TERMS
822 C
823 C 102 CONTINUE
824 SU(IJK)=BT(IJK)+C1*GEN(IJK)*VOL(IJK)*VOL(IJK)*DENSIT*CMU*TE(IJK)
825 SP(IJK)=SP(IJK)+C2*VOL(IJK)*DENSIT*DENSIT*CMO*TE(IJK)/
826 * (VIS(IJK)-VISCOS+SMALL)
827 C
828 C 100 CONTINUE
829 C
830 C-----PROBLEM MODIFICATIONS - BOUNDARY CONDITIONS
831 C
832 IF (IPHI.EQ.1) CALL MODKE(BT)
833 IF (IPHI.EQ.2) CALL MODKE
834 C
835 DO 200 K=2,NKM
836 LKW=LK(K)
837 DO 200 I=2,NIM
838 LIK=LKK+LI(1)
839 DO 200 J=2,NJM
840 IJK=LJK+J
841 AP(IJK)=AE(IJK)+AW(IJK)+AN(IJK)+AS(IJK)+AT(IJK)+AB(IJK)+SP(IJK)
842 AP(IJK)=AP(IJK)*URFPHI
843 SU(IJK)=SU(IJK)+(1.-URFPHI)*AP(IJK)*PHI(IJK)
844 C 200 CONTINUE
845 C
846 C-----SOLVING F.D. EQUATIONS
847 C
848 CALL SOLSIP(PHI,IPHI)
849 C
850 RETURN
851 END
852

```

Oct 12 1996 16:42	sskmod_3d	Page 13
853 C	*****	
854 C	SUBROUTINE SOLSIP(PHI, IPHI)	
855 C	*****	
856 C	*****	
857 C	*****	
858 C	*****	
859 C	*****	
860 C	*****	
861 C	*****	
862 C	*****	
863 C	*****	
864 C	*****	
865 C	*****	
866 C	*****	
867 C	*****	
868 C	*****	
869 C	*****	
870 C	*****	
871 C	*****	
872 C	*****	
873 C	*****	
874 C	*****	
875 C	*****	
876 C	*****	
877 C	*****	
878 C	*****	
879 C	*****	
880 C	*****	
881 C	*****	
882 C	*****	
883 C	*****	
884 C	*****	
885 C	*****	
886 C	*****	
887 C	*****	
888 C	*****	
889 C	*****	
890 C	*****	
891 C	*****	
892 C	*****	
893 C	*****	
894 C	*****	
895 C	*****	
896 C	*****	
897 C	*****	
898 C	*****	
899 C	*****	
900 C	*****	
901 C	*****	
902 C	*****	
903 C	*****	
904 C	*****	
905 C	*****	
906 C	*****	
907 C	*****	
908 C	*****	
909 C	*****	
910 C	*****	
911 C	*****	
912 C	*****	
913 C	*****	
914 C	*****	
915 C	*****	
916 C	*****	
917 C	*****	
918 C	*****	
919 C	*****	
920 C	*****	
921 C	*****	
922 C	*****	
923 C	*****	

Oct 12 1996 16:42	sskmod_3d	Page 14
924 C	*****	
925 C	*****	
926 C	*****	
927 C	*****	
928 C	*****	
929 C	*****	
930 C	*****	
931 C	*****	
932 C	*****	
933 C	*****	
934 C	*****	
935 C	*****	
936 C	*****	
937 C	*****	
938 C	*****	
939 C	*****	
940 C	*****	
941 C	*****	
942 C	*****	
943 C	*****	
944 C	*****	
945 C	*****	
946 C	*****	
947 C	*****	
948 C	*****	
949 C	*****	
950 C	*****	
951 C	*****	
952 C	*****	
953 C	*****	
954 C	*****	
955 C	*****	
956 C	*****	
957 C	*****	
958 C	*****	
959 C	*****	
960 C	*****	
961 C	*****	
962 C	*****	
963 C	*****	
964 C	*****	
965 C	*****	
966 C	*****	
967 C	*****	
968 C	*****	
969 C	*****	
970 C	*****	
971 C	*****	
972 C	*****	
973 C	*****	
974 C	*****	
975 C	*****	
976 C	*****	
977 C	*****	
978 C	*****	
979 C	*****	
980 C	*****	
981 C	*****	
982 C	*****	
983 C	*****	
984 C	*****	
985 C	*****	
986 C	*****	
987 C	*****	
988 C	*****	
989 C	*****	
990 C	*****	

Oct 12 1996 16:42	sskmod_3d	Page 15
991	SP(IJK) = SP(IJK) + AT(IJK)	
992	GO TO 21	
993	C SYMMETRY	
994	31 CONTINUE	
995	GO TO 21	
996	C WALL	
997	41 SU(IJK) = BT(IJK) + GENTT(IJ) *VOL(IJK)	
998	21 AT(IJK)=0.	
999	610 CONTINUE	
1000	C	
1001	C-----SOUTH AND NORTH BOUNDARIES	
1002	C	
1003	DO 620 K=2,NKM	
1004	IK=(I-1)*NK+K	
1005	IJK=LK(K)+LII+2	
1006	GO TO (12,22,32,42) BCFS(IK)	
1007	C INLET	
1008	12 SU(IJK) = SU(IJK) + AS(IJK)*TE(IJK-1)	
1009	SP(IJK) = SP(IJK) + AS(IJK)	
1010	GO TO 22	
1011	C SYMMETRY	
1012	32 CONTINUE	
1013	GO TO 22	
1014	C WALL	
1015	42 SU(IJK) = BT(IJK) + GENTS(IK) *VOL(IJK)	
1016	22 AS(IJK) = 0.	
1017	C	
1018	IJK=LK(K)+LII+NJM	
1019	GO TO (13,23,33,43) BCFN(IK)	
1020	C INLET	
1021	13 SU(IJK) = SU(IJK) + AN(IJK)*TE(IJK+1)	
1022	SP(IJK) = SP(IJK) + AN(IJK)	
1023	GO TO 23	
1024	C SYMMETRY	
1025	33 CONTINUE	
1026	GO TO 23	
1027	C WALL	
1028	43 SU(IJK) = BT(IJK) + GENTN(IK) *VOL(IJK)	
1029	23 AN(IJK)=0.	
1030	620 CONTINUE	
1031	600 CONTINUE	
1032	C	
1033	C-----WEST AND EAST BOUNDARIES	
1034	C	
1035	DO 630 K=2,NKM	
1036	LKK = LK(K)	
1037	KK = (K-1)*NJ	
1038	DO 630 J=2,NJM	
1039	IJK = LKK + LI(2) + J	
1040	JK = KK + J	
1041	GO TO (14,24,34,44) BCFW(JK)	
1042	C	
1043	C INLET	
1044	14 SP(IJK) = SP(IJK) + AW(IJK)	
1045	SU(IJK) = SU(IJK) + AW(IJK)*TE(IJK-NJ)	
1046	GO TO 24	
1047	C SYMMETRY	
1048	34 CONTINUE	
1049	GO TO 24	
1050	C WALL	
1051	44 SU(IJK) = BT(IJK) + GENTW(JK) *VOL(IJK)	
1052	24 AW(IJK)=0.	
1053	C	
1054	IJK = LKK + LI(NIM) + J	

Oct 12 1996 16:42	sskmod_3d	Page 16
1055	GO TO (15,25,35,45) BCPE(JK)	
1056	C INLET	
1057	15 SP(IJK) = SP(IJK) + AE(IJK)	
1058	SU(IJK) = SU(IJK) + AE(IJK)*TE(IJK+NJ)	
1059	GO TO 25	
1060	C SYMMETRY	
1061	35 CONTINUE	
1062	GO TO 25	
1063	C WALL	
1064	45 SU(IJK) = BT(IJK) + GENTE(JK) *VOL(IJK)	
1065	25 AE(IJK)=0.	
1066	C	
1067	630 CONTINUE	
1068	RETURN	
1069	END	
1070	C	
1071	C-----	
1072	SUBROUTINE MODD	
1073	C-----	
1074	INCLUDE 'kmod.h'	
1075	C	
1076	C-----DISSIPATION OF TURB. EN. BOUNDARY CONDITIONS	
1077	DATA CAPPA/0.4197/	
1078	CMU25=SQRT(SQRT(CMU))	
1079	CMU75=CMU25**3	
1080	C	
1081	CONST=GREAT*CMU75/CAPPA	
1082	C	
1083	C-----BOTTOM AND TOP BOUNDARIES	
1084	C	
1085	DO 600 I=2,NIM	
1086	LII = LI(I)	
1087	DO 610 J=2,NJM	
1088	IJ = LII+J	
1089	IJK = LK(2) + IJ	
1090	GO TO (10,20,30,40) BCFB(IJ)	
1091	C INLET	
1092	10 SU(IJK) = SU(IJK) + AB(IJK)*ED(IJK-NIJ)	
1093	SP(IJK) = SP(IJK) + AB(IJK)	
1094	GO TO 20	
1095	C SYMMETRY	
1096	30 CONTINUE	
1097	GO TO 20	
1098	C WALL	
1099	40 TE(IJK) = AMAX1(TE(IJK),-TE(IJK))	
1100	SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNB(IJ)	
1101	SP(IJK) = GREAT	
1102	20 AB(IJK)=0.	
1103	C	
1104	IJK=LK(NKM)+IJ	
1105	GO TO (11,21,31,41) BCFT(IJ)	
1106	C INLET	
1107	11 SU(IJK) = SU(IJK) + AT(IJK)*ED(IJK-NIJ)	
1108	SP(IJK) = SP(IJK) + AT(IJK)	
1109	GO TO 21	
1110	C SYMMETRY	
1111	31 CONTINUE	
1112	GO TO 21	
1113	C WALL	
1114	41 TE(IJK) = AMAX1(TE(IJK),-TE(IJK))	
1115	SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNT(IJ)	
1116	SP(IJK) = GREAT	

Oct 12 1996 16:42	sskmod_3d	Page 17
1117 21	AT(IJK)=0.	
1118 610	CONTINUE	
1119 C	DO 60 K=2,NJM	
1120 C	IK=(I-1)*NK+K	
1121 C	IK=LK(K)+LII+2	
1122 C	GO TO (12,22,32,42) BCFS(IK)	
1123 C	INLET	
1124 12	SU(IJK) = SU(IJK) + AS(IJK)*ED(IJK-1)	
1125 12	SP(IJK) = SP(IJK) + AS(IJK)	
1126 C	GO TO 22	
1127 C	SYMMETRY	
1128 32	CONTINUE	
1129 C	GO TO 22	
1130 C	WALL	
1131 42	TE(IJK) = AMAX1(TE(IJK), -TE(IJK))	
1132 C		
1133 C	SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNS(IK)	
1134 22	SP(IJK) = GREAT	
1135 22	AS(IJK) = 0.	
1136 C	IK=LK(K)+LII+NJM	
1137 C	GO TO (13,23,33,43) BCFN(IK)	
1138 C	INLET	
1139 13	SU(IJK) = SU(IJK) + AN(IJK)*ED(IJK+1)	
1140 13	SP(IJK) = SP(IJK) + AN(IJK)	
1141 C	GO TO 23	
1142 C	SYMMETRY	
1143 33	CONTINUE	
1144 C	GO TO 23	
1145 C	WALL	
1146 43	TE(IJK) = AMAX1(TE(IJK), -TE(IJK))	
1147 C		
1148 C	SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNN(IK)	
1149 23	AN(IJK)=0.	
1150 600	CONTINUE	
1151 C	WEST AND EAST BOUNDARIES	
1152 C	DO 630 K=2,NKM	
1153 C	LKK = LK(K)	
1154 C	KK = (K-1)*NJ	
1155 C	DO 630 J=2,NJM	
1156 C	IJK = LKK + LI(2) + J	
1157 C	JK = KK + J	
1158 C	GO TO (14,24,34,44) BCFW(IJK)	
1159 C	INLET	
1160 14	SP(IJK) = SP(IJK) + AW(IJK)	
1161 C	SU(IJK) = SU(IJK) + AW(IJK)*ED(IJK-NJ)	
1162 C	GO TO 24	
1163 C	SYMMETRY	
1164 34	CONTINUE	
1165 C	GO TO 24	
1166 C	WALL	
1167 44	TE(IJK) = AMAX1(TE(IJK), -TE(IJK))	
1168 C		
1169 C	SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNW(IK)	
1170 24	AW(IJK) = GREAT	
1171 C		
1172 C		
1173 C		

Oct 12 1996 16:42	sskmod_3d	Page 18
1178	IJK = LKK + LI(NIM) + J	
1179	GO TO (15,25,35,45) BCFE(IJK)	
1180 C	INLET	
1181 15	SP(IJK) = SP(IJK) + AE(IJK)	
1182	SU(IJK) = SU(IJK) + AE(IJK)*ED(IJK+NJ)	
1183	GO TO 25	
1184 C	SYMMETRY	
1185 35	CONTINUE	
1186	GO TO 25	
1187 C	WALL	
1188 45	TE(IJK) = AMAX1(TE(IJK), -TE(IJK))	
1189	SU(IJK) = CONST*SQRT(TE(IJK))*TE(IJK)/DNE(IJK)	
1190	SP(IJK) = GREAT	
1191 25	AE(IJK)=0.	
1192 C	630 CONTINUE	
1193	RETURN	
1194	END	
1195 C		
1196 C	SUBROUTINE MODIFY (SU,BP)	
1197 C	INCLUDE 'kmod.h'	
1198 C	DIMENSION SU(JXYZ),BP(JXYZ)	
1199	DATA CMU25,CAPPA,ELOG/0.5477,0.4197,9.0/	
1200	-----BOTTOM BOUNDARY (WALL)	
1201	DO I=2,NIM	
1202	LII=LI(I)	
1203	DO J=2,NJM	
1204	IJ=LII+J	
1205	IJK=LK(2)+IJ	
1206	IMJKN=IJK-NJ	
1207	IF(BCFB(IJ).EQ.4) THEN	
1208	DX1=X(IJKN-1)-X(IMJKN)	
1209	DY1=Y(IJKN-1)-Y(IMJKN)	
1210	DZ1=Z(IJKN-1)-Z(IMJKN)	
1211	DX2=X(IJKN)-X(IMJKN-1)	
1212	DY2=Y(IJKN)-Y(IMJKN-1)	
1213	DZ2=Z(IJKN)-Z(IMJKN-1)	
1214	LW=IJK	
1215	LB=IJK-NIJ	
1216	DELN=DNB(IJ)	
1217	CALL WALLFBN (LW,LB,DX1,DX2,DY1,DY2,DZ1,DZ2,CAPPA,DEN,GENTE,DELN,	
1218	& CMU25,VISC,ELOG,YPLS,SPU,SUV,SPW,SUW,TAU)	
1219	BP(IJK)=BP(IJK)+SPU	
1220	SU(IJK)=SU(IJK)+SUV	
1221	SPWB(IJ)=SPW	
1222	SUVB(IJ)=SUV	
1223	SUNB(IJ)=SUW	
1224	GENTB(IJ)=GENTE	
1225	AB(IJK)=0.	
1226	ENDIF	
1227	ENDDO	
1228	-----TOP BOUNDARY	
1229	DO I=2,NIM	
1230	LII = LI(I)	
1231	DO J=2,NJM	
1232	IJ=LII+J	
1233	IJK=LK(NJM)+IJ	
1234	IMJKN=IJK-NJ	
1235	IF(BCFT(IJ).EQ.4) THEN	
1236	DX1=X(IJK)-X(IMJKN-1)	
1237		
1238		
1239		
1240		
1241		
1242		
1243		
1244		

Oct 12 1996 16:42

sskmod_3d

Page 19

```

1245 DY1=Y(IJK)-Y(IMJK-1)
1246 DZ1=Z(IJK)-Z(IMJK-1)
1247 DX2=X(IJK-1)-X(IMJK)
1248 DY2=Y(IJK-1)-Y(IMJK)
1249 DZ2=Z(IJK-1)-Z(IMJK)
1250 LW=IJK
1251 LB=IJK+NIJ
1252 DELN=DNT(IJ)
1253 CALL WALLFN (LW, LB, DX1, DX2, DY1, DY2, DZ1, DZ2, CAPA, DEN, GENTE, DELN,
1254 & CMU25, VISC, ELOG, YPLS, SPU, SUV, SPW, SUMW, TAU)
1255 BP(IJK)=BP(IJK)+SPU
1256 SU(IJK)=SU(IJK)+SUV
1257 SPVT(IJ)=SPV
1258 SPWT(IJ)=SPW
1259 SUVT(IJ)=SUV
1260 SUMT(IJ)=SUMW
1261 GENTT(IJ)=GENTE
1262 AT(IJK)=0.0
1263 ENDDIF
1264 ENDDO
1265 C-----SOUTH BOUNDARY
1266 DO I=2, NIM
1267 II = (I-1)*NK
1268 DO K=2, NKM
1269 IK = II + K
1270 IJK = LK(K) + LI(I) + 2
1271 IJKN = IJK - NIJ
1272 IMJK = IJK - NJ
1273 IMJKN = IJKN - NJ
1274 IJMK = IJK - 1
1275 IF(BCFS(IK), EQ. 4) THEN
1276 DX1 = X(IMJK-1) - X(IJKN-1)
1277 DY1 = Y(IMJK-1) - Y(IJKN-1)
1278 DZ1 = Z(IMJK-1) - Z(IJKN-1)
1279 DX2 = X(IJK-1) - X(IMJKN-1)
1280 DY2 = Y(IJK-1) - Y(IMJKN-1)
1281 DZ2 = Z(IJK-1) - Z(IMJKN-1)
1282 LW = IJK
1283 LB = IJK - 1
1284 DELN = DNS(IK)
1285 CALL WALLFN (LW, LB, DX1, DX2, DY1, DY2, DZ1, DZ2, CAPA, DEN, GENTE, DELN,
1286 & CMU25, VISC, ELOG, YPLS, SPU, SUV, SPW, SUMW, TAU)
1287 BP(IJK) = BP(IJK) + SPU
1288 SU(IJK) = SU(IJK) + SUV
1289 SPVS(IK) = SPV
1290 SPWS(IK) = SPW
1291 SUV(S(IK) = SUV
1292 SUMS(IK) = SUM
1293 GENTS(IK) = GENTE
1294 AS(IJK)=0.0
1295 ENDDIF
1296 ENDDO
1297 C-----NORTH BOUNDARY
1298 DO I=2, NIM
1299 II = (I-1)*NK + K
1300 IJKN = LK(K) + LI(I) + NJM
1301 IMJK = IJKN - NIJ
1302 IMJKN = IJKN - NJ
1303 IJPK = IJK + 1
1304 IF(BCFN(IK), EQ. 4) THEN
1305 DX1 = X(IJK) - X(IMJKN)
1306 DY1 = Y(IJK) - Y(IMJKN)
1307 DZ1 = Z(IJK) - Z(IMJKN)
1308 DX2 = X(IMJK) - X(IJKN)
1309 DY2 = Y(IMJK) - Y(IJKN)
1310 DZ2 = Z(IMJK) - Z(IJKN)
1311 LW = IJK
1312
1313
1314
1315

```

Oct 12 1996 16:42

sskmod_3d

Page 20

```

1316 LB = IJK + 1
1317 DELN = DNN(IK)
1318 CALL WALLFN (LW, LB, DX1, DX2, DY1, DY2, DZ1, DZ2, CAPA, DEN, GENTE, DELN,
1319 & CMU25, VISC, ELOG, YPLS, SPU, SUV, SPW, SUMW, TAU)
1320 BP(IJK) = BP(IJK) + SPU
1321 SU(IJK) = SU(IJK) + SUV
1322 SPVN(IK) = SPV
1323 SPVN(IK) = SPW
1324 SUVN(IK) = SUV
1325 SUMN(IK) = SUM
1326 GENTN(IK) = GENT
1327 AN(IJK) = 0.0
1328 ENDDIF
1329 ENDDO
1330 C-----WEST BOUNDARY
1331 DO K=2, NKM
1332 LKK = LK(K)
1333 KK = (K-1)*NJ
1334 DO J=2, NJM
1335 JK = KK + J
1336 IJK = LKK + LI(2) + J
1337 IMJK = IJK - NJ
1338 IJMK = IJK - 1
1339 IJKN = IJK - NIJ
1340 IJMKM = IJKN - 1
1341 IF(BCFW(IK), EQ. 4) THEN
1342 DX1 = X(IJMK-NJ) - X(IJKN-NJ)
1343 DY1 = Y(IJMK-NJ) - Y(IJKN-NJ)
1344 DZ1 = Z(IJMK-NJ) - Z(IJKN-NJ)
1345 DX2 = X(IJKN-NJ) - X(IJMKM-NJ)
1346 DY2 = Y(IJKN-NJ) - Y(IJMKM-NJ)
1347 DZ2 = Z(IJKN-NJ) - Z(IJMKM-NJ)
1348 LW = IJK
1349 LB = IJK - NJ
1350 DELN = DNN(IK)
1351 CALL WALLFN (LW, LB, DX1, DX2, DY1, DY2, DZ1, DZ2, CAPA, DEN, GENTE, DELN,
1352 & CMU25, VISC, ELOG, YPLS, SPU, SUV, SPW, SUMW, TAU)
1353 BP(IJK) = BP(IJK) + SPU
1354 SU(IJK) = SU(IJK) + SUV
1355 SPVM(IK) = SPV
1356 SPVM(IK) = SPW
1357 SPVM(IK) = SUV
1358 SPVM(IK) = SPW
1359 SUMV(IK) = SUM
1360 GENTV(IK) = GENT
1361 AW(IJK)=0.0
1362 ENDDIF
1363 ENDDO
1364 C-----EAST BOUNDARY
1365 DO K=2, NKM
1366 LKK = LK(K)
1367 KK = (K-1)*NJ
1368 DO J=2, NJM
1369 JK = KK + J
1370 IJK = LKK + LI(NIM) + J
1371 IJKN = IJK - NIJ
1372 IJMK = IJK - 1
1373 IJMKM = IJMK - NIJ
1374 IFJK = IJK + NJ
1375 IF(BCFE(IK), EQ. 4) THEN
1376 DX1 = X(IJK) - X(IJMKM)
1377 DY1 = Y(IJK) - Y(IJMKM)
1378 DZ1 = Z(IJK) - Z(IJMKM)
1379 DX2 = X(IJMK) - X(IJKB)
1380 DY2 = Y(IJMK) - Y(IJKB)
1381 DZ2 = Z(IJMK) - Z(IJKB)
1382 LW = IJK
1383 LB = IJK + NJ
1384 DELN = DNE(IK)
1385 CALL WALLFN (LW, LB, DX1, DX2, DY1, DY2, DZ1, DZ2, CAPA, DEN, GENTE, DELN,
1386

```

Oct 12 1996 16:42	sskmod_3d	Page 21
1387	& CMU25,VISC,ELOG,YPLS,SPU,SUV,SPV,SUV,SPW,SUWV,TAU)	
1388	BP(IJK) = BP(IJK) + SPU	
1389	SU(IJK) = SU(IJK) + SUV	
1390	SPVE(IJK) = SPV	
1391	SUVE(IJK) = SUV	
1392	SPWE(IJK) = SPW	
1393	SUWE(IJK) = SUW	
1394	GENTE(IJK) = GENT	
1395	AE(IJK)=0.0	
1396	ENDIF	
1397	ENDDO	
1398	ENDDO	
1399	C	
1400	C-----	
1401	SUBROUTINE WALLFN(LW, LB, DX1, DX2, DY1, DY2, DZ1, DZ2, CAPA,	
1402	& DEN, GENTE, DELN, CMU25, VISC, ELOG, YPLS, SPV, SUU,	
1403	& SPV, SUV, SPW, SUWV, TAU)	
1404	C-----	
1405	INCLUDE 'kmod.h'	
1406	WALL CELL FACE AREA	
1407	XAN=DY1*DZ2-DY2*DZ1	
1408	YAN=DX2*DZ1-DX1*DZ2	
1409	ZAN=DX1*DY2-DX2*DY1	
1410	ARN=0.5*SORT(XAN**2+YAN**2+ZAN**2)	
1411	ARNR=0.5/ARN	
1412	C-----COMPONENTS OF UNIT NORMAL VECTOR	
1413	ALFAN=XAN*ARNR	
1414	BETAN=YAN*ARNR	
1415	GAMAN=ZAN*ARNR	
1416	C-----CALCULATE Y+ AND LAMBDA-WALL COEFF.	
1417	CONST=DN*CMU25*SORT(TE(LW))	
1418	YPLS=DELN*CONST/(VISC+1.E-30)	
1419	TCOEFF=VISC/DELN	
1420	IF(YPLS.GT.11.63) TCOEFF=CONST*CAPA/(ALOG(ELOG*YPLS))	
1421	TAR=TCOEFF*ARN	
1422	C-----SOURCE TERMS FOR VELOCITIES	
1423	SPU=TAR*(1.-ALFAN**2)	
1424	SPV=TAR*(1.-BETAN**2)	
1425	SPW=TAR*(1.-GAMAN**2)	
1426	SUV=TAR*ALFAN*(BETAN*V(LW)+GAMAN*W(LW))	
1427	SUW=TAR*BETAN*(ALFAN*U(LW)+GAMAN*W(LW))	
1428	SUWV=TAR*GAMAN*(ALFAN*U(LW)+BETAN*V(LW))	
1429	C-----VELOCITY PARALLEL TO WALL	
1430	UP=SQRT((SPU*U(LW)-SUV)**2+(SPV*V(LW)-SUW)**2+ * (SPW*W(LW)-SUWV)**2)/(TAR+1.E-30)	
1431	UP=ABS(UP-SORT(U(LB)*2+V(LB)*2+W(LB)*2))	
1432	C-----WALL SHEAR STRESS AND GENER. TERM	
1433	SUV=SUV+TAR*U(LB)	
1434	SUW=SUV+TAR*V(LB)	
1435	SUWV=SUV+TAR*W(LB)	
1436	TAU=UP*TCOEFF	
1437	GENTE=TAU*UP/DELN	
1438	RETURN	
1439	END	
1440	C	
1441	C-----	
1442	C-----	
1443	C-----	
1444	C-- kmod.h	
1445	C	
1446	PARAMETER (JX=70)	
1447	PARAMETER (JY=41)	
1448	PARAMETER (JZ=41)	
1449	PARAMETER (JXY=JX*JY)	
1450	PARAMETER (JXZ=JX*JZ)	
1451	PARAMETER (JYZ=JY*JZ)	
1452	PARAMETER (JXYZ=JX*JY*JZ)	
1453	C	
1454	COMMON NI, NJ, NK, NIM, NKM, NIJ, NIK, NIJK, LK(JZ), LI(JX), IU, IV, IW,	
1455	* IP, ITE, IED, IVIS, IEN, NITER, MAXIT, ITSTEP, ITEST, ICAI(10),	

Oct 12 1996 16:42	sskmod_3d	Page 22
1456	* IMON, JMON, KMOM, IJMKOM, IJMKPR, IDIR, NSWP(2), G, SOR(2)	
1457	,X(JXYZ),Y(JXYZ),Z(JXYZ),FX(JXYZ),FY(JXYZ),FZ(JXYZ),	
1458	VOL(JXYZ),DP1(JXYZ),DP2(JXYZ),DP3(JXYZ)	
1459	,AE(JXYZ),AW(JXYZ),AN(JXYZ),AS(JXYZ),AT(JXYZ),AB(JXYZ),	
1460	AP(JXYZ),SU(JXYZ),BE(JXYZ),BN(JXYZ),BT(JXYZ),RES(JXYZ),	
1461	SWU(JXYZ),DB(JXY),DW(JY)	
1462	,RESOR(2),SNORIN(2),PRTNV(2),URF(2),HAF,QTR,AHT,OMEGA	
1463	,ALFA,DENSIT,SORMAX,VISCOS,READI,WRIT7,IOBST,GREAT,SMALL,	
1464	C1,C2,CAPPA,CHU,CMU25,CMU75,SUM(JY),SUB(JXY),UWB(JY),	
1465	VWB(JY),WMB(JY),FZBB(JXY),FXMW(JY)	
1466	COMMON /UWV/ P(JXYZ),PP(JXYZ),VIS(JXYZ),U(JXYZ),V(JXYZ),	
1467	* W(JXYZ),TE(JXYZ),ED(JXYZ),T(1),F1(JXYZ),F2(JXYZ),F3(JXYZ)	
1468		

CHAPTER 7

3D Algebraic Stress Turbulence Model

Table of Contents

	page
7.1 Introduction	91
7.2 Theory and Model Equations	91
References	94
Appendix F	95

7.1 Introduction

In this section a description is given of the three-dimensional Algebraic Stress turbulence Model (ASM) based on the work of Rodi [1]. The model is coded as a self contained computer program to compute turbulent flow quantities when interfaced with a CFD solver. Detailed description of the module structure, variables used and how to interface the module with CFD flow solvers are given in the Appendix.

The module uses as input the mean flow properties, as computed by conventional CFD solvers, and calculates the Reynolds stresses, turbulent kinetic energy and the energy dissipation. It is structured to be self-contained and compatible with many CFD codes. The module has not been tested thoroughly due to the ending of the contract earlier than scheduled. Some testing of the module has been done at UAH but that also has been put on hold. However, the module as assembled is capable of interfacing with a number CFD solvers.

The module computes turbulent flow quantities in three-dimensional body-fitted geometry with or without rotation about any one of the three axis. The standard wall functions is used for the near wall treatment.

7.2 Theory and Model Equations

The Algebraic Stress (ASM) module discussed here is based on the work of Rodi [1]. The idea is to simplify or truncate the Reynolds stress equation by approximating the convective and diffusive transport of the Reynolds stresses $\overline{u_i u_j}$ in terms of the corresponding transport of turbulent energy. This allows the transport equation for the stresses to be expressed as a set of algebraic formulae containing the turbulence energy and its rate of dissipation as unknowns in the form:

$$\overline{u_i u_j} = \frac{k}{(P-\varepsilon)} \left[P_{ij} - \frac{2}{3} \delta_{ij} \varepsilon + \Phi_{ij} \right]$$

where P_{ij} = Production and $P = \frac{1}{2} P_{kk}$

Φ_{ij} = pressure-strain redistribution

$$\Phi_{ij} = \Phi_{ij,1} + \Phi_{ij,2} + \Phi_{ij,1w} + \Phi_{ij,2w}$$

Rotta's linear return-to-isotropy concept for the non-linear part

$$\Phi_{ij,1} = -C_1 \frac{\varepsilon}{k} (\overline{u_i u_j} - \frac{2}{3} k \delta_{ij})$$

is used and the "isotropization of production" concept for the linear "rapid" part

$$\Phi_{ij,2} = -C_2 (P_{ij} - \frac{2}{3} P \delta_{ij})$$

is used. Gibson and Launder [2] concept for the wall reflection terms is used as

$$\Phi_{ij,1w} = C_{1w} \rho \frac{\varepsilon}{k} (\overline{u_k u_m} n_k n_m \delta_{ij} - \frac{3}{2} \overline{u_k u_i} n_k n_j - \frac{3}{2} \overline{u_k u_j} n_k n_i) f$$

$$\Phi_{ij,2w} = C_{2w} (\Phi_{km,2} n_k n_m \delta_{ij} - \frac{3}{2} \Phi_{ik,2} n_k n_j - \frac{3}{2} \Phi_{jk,2} n_k n_i) f$$

where (n_i) is the wall-normal unit vector in the i -direction. The wall-distance function (f) represents the ratio of the turbulence length scale ($L_\varepsilon = \frac{k^{3/2}}{\varepsilon}$) and the wall distance and is given

as

$$f = (\frac{C_m^{0.75} k^{1.5}}{K \varepsilon}) \frac{1}{\Delta n}$$

with Δn being the wall-normal distance.

The resulting set of algebraic equations for the Reynolds stresses can be arranged in the form

$$A_{ij} \overline{u^2} + B_{ij} \overline{v^2} + C_{ij} \overline{w^2} + D_{ij} \overline{uv} + E_{ij} \overline{vw} + F_{ij} \overline{uw} = G_{ij}$$

where A_{ij} , B_{ij} , C_{ij} , D_{ij} , E_{ij} , F_{ij} , and G_{ij} are functions of the mean and turbulent flow variables.

The above equation can be solved iteratively in the main flow solver. However, the algebraic system of equations is stiff and convergence difficulties are encountered when solved iteratively. Therefore, the set of equations was cast in the general matrix form $\underline{A} \underline{T} = \underline{B}$, where

$$\begin{aligned}
\underline{\mathbf{A}} = & \begin{bmatrix} \frac{3\varepsilon}{2\lambda k} + 2\frac{\partial U}{\partial x} & -\frac{\partial V}{\partial y} & -\frac{\partial W}{\partial z} & 2\frac{\partial U}{\partial y} - \frac{\partial V}{\partial x} - 6C_0\Omega_z & -(\frac{\partial V}{\partial z} + \frac{\partial W}{\partial y}) & 2\frac{\partial U}{\partial z} - \frac{\partial W}{\partial x} + 6C_0\Omega_y \\ -\frac{\partial U}{\partial x} & \frac{3\varepsilon}{2\lambda k} + 2\frac{\partial V}{\partial y} & -\frac{\partial W}{\partial z} & 2\frac{\partial V}{\partial x} - \frac{\partial U}{\partial y} + 6C_0\Omega_z & 2\frac{\partial V}{\partial z} - \frac{\partial W}{\partial y} - 6C_0\Omega_x & -(\frac{\partial U}{\partial z} + \frac{\partial W}{\partial x}) \\ -\frac{\partial U}{\partial x} & -\frac{\partial V}{\partial y} & \frac{3\varepsilon}{2\lambda k} + 2\frac{\partial W}{\partial z} & -(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}) & -\frac{\partial V}{\partial z} + 2\frac{\partial W}{\partial y} + 6C_0\Omega_x & -\frac{\partial U}{\partial z} + 2\frac{\partial W}{\partial x} - 6C_0\Omega_y \\ \frac{\partial V}{\partial x} + 2C_0\Omega_z & \frac{\partial U}{\partial y} - 2C_0\Omega_z & 0 & \frac{\varepsilon}{\lambda k} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} & \frac{\partial U}{\partial z} + 2C_0\Omega_y & \frac{\partial V}{\partial z} - 2C_0\Omega_x \\ 0 & \frac{\partial W}{\partial y} + 2C_0\Omega_x & \frac{\partial V}{\partial z} - 2C_0\Omega_x & \frac{\partial W}{\partial x} - 2C_0\Omega_y & \frac{\varepsilon}{\lambda k} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} & \frac{\partial V}{\partial x} + 2C_0\Omega_z \\ \frac{\partial W}{\partial x} - 2C_0\Omega_y & 0 & \frac{\partial U}{\partial z} + 2C_0\Omega_y & \frac{\partial W}{\partial y} + 2C_0\Omega_x & \frac{\partial U}{\partial y} + 2C_0\Omega_z & \frac{\varepsilon}{\lambda k} + \frac{\partial U}{\partial x} + \frac{\partial W}{\partial z} \end{bmatrix}
\end{aligned}$$

$$\underline{\mathbf{T}} = [\rho \overline{u u}, \rho \overline{v v}, \rho \overline{w w}, \rho \overline{u v}, \rho \overline{v w}, \rho \overline{u w}]^T$$

$$\begin{aligned}
\underline{\mathbf{B}} = & \begin{bmatrix} \frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)} (\Phi_{11,1w} + \Phi_{11,2w}) \\ \frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)} (\Phi_{22,1w} + \Phi_{22,2w}) \\ \frac{\rho\varepsilon}{\lambda} + \frac{3}{2(1-C_2)} (\Phi_{33,1w} + \Phi_{33,2w}) \\ \frac{1}{(1-C_2)} (\Phi_{12,1w} + \Phi_{12,2w}) \\ \frac{1}{(1-C_2)} (\Phi_{23,1w} + \Phi_{23,2w}) \\ \frac{1}{(1-C_2)} (\Phi_{13,1w} + \Phi_{13,2w}) \end{bmatrix}
\end{aligned}$$

$$\text{where } \lambda = \frac{1-C_2}{C_1 - 1 + \frac{P}{\rho\varepsilon}}$$

The matrix was inverted at each iteration step to obtain a converged solution.

REFERENCES

- [1] Rodi, W., 'A New Algebraic Stress Relations for Calculating the Reynolds Stresses'
Z. Ang. Math. und Mech., vol. 56, pp. 219-, 1976.

- [4] Gibson M. and Launder B., 'Ground Effects on Pressure Fluctuations in the
Atmosphere boundary Layer', J. Fluid Mech. vol. 86, pt. 3, pp. 491-., 1978.

APPENDIX F

3D Algebraic Stress Module Deck

ASMOD is a FORTRAN source code to solve 2D/Axisymmetric turbulent flow quantities using the algebraic stress model when interfaced with a main flow solver. The module consists of the main routine ASMOD that calls a number of subroutines to perform different functions that will be explained below.

Subroutine ASMMOD

This is basically the main routine that reads through its argument list different variables from the calling flow solver which are described below.

List of Argument Variable Names

INITASM	Initialization parameter that writes and sets variables
NIM	Number of grid nodes in the i (or x) direction
NJM	Number of grid nodes in the j (or y) direction
NKM	Number of grid nodes in the k (or z) direction
LI	$LI(I)=(I-1)*NJ$, dimensioned to NX . Calculated as in subroutine GRIDG of the 3D $k-\epsilon$ module.
LK	$LK(K)=(K-1)*NI*NJ$ dimensioned to NZ . Calculated as in subroutine GRIDG of the 3D $k-\epsilon$ module.
FX	grid interpolation factor in the x-direction
FY	grid interpolation factor in the y-direction
FZ	grid interpolation factor in the z-direction
X	Grid node locations in the x or ξ -direction, dimensioned to $X(JXYZ=NX*NY*NZ)$
Y	Grid node locations in the y or η -direction, dimensioned to $Y(JXYZ)$
Z	Grid node locations in the z or ζ -direction, dimensioned to $Z(JXYZ)$
VOL	Control cell volume (similar to that calculated in GRIDG of $k-\epsilon$ module)
U	mean velocity in x or ξ -direction, dimensioned to $U(JXYZ)$ (input from the flow solver)

V	Mean velocity in the y or η -direction, dimensioned yo V(JXYZ) (input from the flow solver)
W	Mean velocity in the z or ζ -direction, dimensioned yo W(JXYZ) (input from the flow solver)
VIS	Eddy viscosity
TE	Turbulent kinetic energy, dimensioned to TE(JXYZ) calculated in the module.
ED	Turbulent energy dissipation, dimensioned to ED(JXYZ)
U2	Normal Reynolds stress component $\overline{u^2}$, calculated in the module
V2	Normal Reynolds stress component $\overline{v^2}$, calculated in the module
W2	Normal Reynolds stress component $\overline{w^2}$, calculated in the module
UV	Shear stress component \overline{uv} , calculated in the module
VW	Shear stress component \overline{vw} ,calculated in the module
UW	Shear stress component \overline{uw} , calculated in the module
GEN	Turbulent energy generation term
SUASM	Source term for the U-momentum equation due to Reynolds stress gradients. Calculated in the module and passed to the main solver.
SVASM	Source term for the V-momentum equation due to Reynolds stress gradients. Calculated in the module and passed to the main solver.
SWASM	Source term for the W-momentum equation due to Reynolds stress gradients. Calculated in the module and passed to the main solver.
BCFW	Boundary condition flag along the west boundary (or y-z plane). It must have one for each boundary node set to; 1-inlet, 2-outlet, 3-symmetry and 4-wall. For example for an outlet flow condition on the west boundary set BCFW to (NY*NZ)*2, and similarly for the other boundaries, dimensioned to BCFW(JYZ=NY*NZ) (input from flow solver)
BCFE	Boundary condition flag for the east boundary dimensioned to BCFE(JYZ)
BCFS	Boundary condition flag for the south boundary dimensioned to BCFS(JXZ)
BCFN	Boundary condition flag for the north boundary dimensioned to BCFN(JYZ)
BCFB	Boundary condition flag for the bottom boundary dimensioned to BCFB(JXY)
BCFT	Boundary condition flag for the top boundary dimensioned to BCFT(JXY)
GENTW	Turbulent generation terms calculated from the wall functions close to the wall in the west direction. Similarly for the other GENTE, GENTS....
OMX	Frame rotation term in the x-direction. Similarly OMY & OMZ in the y and z-directions respectively

DENSIT	Constant density
VISCOS	Kinematic viscosity

All dimensions considered are one-dimensional. The position of any node is defined as $IJK = (I,J,K) = (I-1)*NJ + (K-1)*NJ + J$, where NI, NJ and NK are the number of grid nodes in the X, Y and Z-directions respectively. It is assumed that grid related data such as control volumes and interpolation factors be passed to the module from an external grid generator, similar to the one listed in the 3D k-e module (Chapter 6).

Subroutine CALPIJ

This subroutine calculates the production terms of the individual stress components.

Subroutine CALUIIJ

This subroutine calculates the individual stress component from its algebraic equation. It sets the coefficients of the algebraic stress equations which are solved implicitly at each iteration step by inverting a 6x6 matrix.

Subroutine SORUVW

This subroutine calculates the source terms needed in the momentum equation of the main CFD solver due to Reynolds stress gradients.

Subroutine SOLV

This subroutine is a Gaussian elimination solver to invert a 6x6 matrices.

Subroutine WALSTRS

This subroutine calculates the Reynolds stresses near the walls based on wall functions.

Oct 12 1996 16:37 asmod_3d Page 3

```

143 LJK = LI(1) + LK(K)
144 DO 20 J=2,NJM
145 C
146 LJK = LK(K) + J
147 IPJK = LJK + NJ
148 IMJK = LJK - NJ
149 IKJP = LJK + NIJ
150 IJPK = LJK - NIJ
151 C
152 C
153 DUE(IJK)=U2(IPJK)-U2(IJK)
154 DUJ(IJK)=U2(IJPK)-U2(IJK)
155 DUUT(IJK)=U2(IJPK)-U2(IJK)
156 DVE(IJK)=V2(IPJK)-V2(IJK)
157 DVN(IJK)=V2(IJPK)-V2(IJK)
158 DVUT(IJK)=V2(IJPK)-V2(IJK)
159 DWE(IJK)=W2(IPJK)-W2(IJK)
160 DWN(IJK)=W2(IJPK)-W2(IJK)
161 DWT(IJK)=W2(IJPK)-W2(IJK)
162 C
163 DUU1(IJK) = 0.5*(U2(IPJK) - U2(IMJK))
164 DVU1(IJK) = 0.5*(V2(IPJK) - V2(IMJK))
165 DWU1(IJK) = 0.5*(W2(IPJK) - W2(IMJK))
166 DUU1(IJK) = 0.5*(U2(IPJK) - U2(IMJK))
167 DUU1(IJK) = 0.5*(U2(IPJK) - U2(IMJK))
168 DWU1(IJK) = 0.5*(W2(IPJK) - W2(IMJK))
169 C
170 DUU2(IJK) = 0.5*(U2(IJPK+1)-U2(IJK-1))
171 DVU2(IJK) = 0.5*(V2(IJPK+1)-V2(IJK-1))
172 DWU2(IJK) = 0.5*(W2(IJPK+1)-W2(IJK-1))
173 DUU2(IJK) = 0.5*(U2(IJPK+1)-U2(IJK-1))
174 DVU2(IJK) = 0.5*(V2(IJPK+1)-V2(IJK-1))
175 DWU2(IJK) = 0.5*(W2(IJPK+1)-W2(IJK-1))
176 C
177 DUU3(IJK) = 0.5*(U2(IJPK) - U2(IJPK))
178 DVU3(IJK) = 0.5*(V2(IJPK) - V2(IJPK))
179 DWU3(IJK) = 0.5*(W2(IJPK) - W2(IJPK))
180 DUU3(IJK) = 0.5*(U2(IJPK) - U2(IJPK))
181 DVU3(IJK) = 0.5*(V2(IJPK) - V2(IJPK))
182 DWU3(IJK) = 0.5*(W2(IJPK) - W2(IJPK))
183 C
184 C
185 DO 31 K=2,NKM
186 C ALONG WEST AND EAST BOUNDARY
187 KK = (K-1)*NJ
188 DO 32 J=2,NJM
189 JK = KK + J
190 IJK = LK(K) + LI(2) + J
191 IPJK = IJK + NJ
192 IPPJK = IPJK + NJ
193 IF(BCFW(IJK).NE.6) THEN
194 DUU1(IJK) = 0.5*(-3.0*U2(IJK) + 4.0*U2(IPJK) - U2(IPPK))
195 DVU1(IJK) = 0.5*(-3.0*V2(IJK) + 4.0*V2(IPJK) - V2(IPPK))
196 DWU1(IJK) = 0.5*(-3.0*W2(IJK) + 4.0*W2(IPJK) - W2(IPPK))
197 DUU1(IJK) = 0.5*(-3.0*U2(IJK) + 4.0*U2(IPJK) - U2(IPPK))
198 DVU1(IJK) = 0.5*(-3.0*V2(IJK) + 4.0*V2(IPJK) - V2(IPPK))
199 DWU1(IJK) = 0.5*(-3.0*W2(IJK) + 4.0*W2(IPJK) - W2(IPPK))
200 ENDIF
201 IJK = LK(K) + LI(NIM) + J
202 IMJK = IJK - NJ
203 IMWJK = IMJK - NJ
204 IF(BCFE(IJK).NE.6) THEN
205 DUU1(IJK) = 0.5*(U2(IMWJK) + 3.0*U2(IJK))
206 DVU1(IJK) = 0.5*(V2(IMWJK) + 3.0*V2(IJK))
207 DWU1(IJK) = 0.5*(W2(IMWJK) + 3.0*W2(IJK))
208 DUU1(IJK) = 0.5*(U2(IMWJK) + 3.0*U2(IJK))
209 DVU1(IJK) = 0.5*(V2(IMWJK) + 3.0*V2(IJK))
210 DWU1(IJK) = 0.5*(W2(IMWJK) + 3.0*W2(IJK))
211 ENDIF
212 C
213 C

```

Oct 12 1996 16:37 asmod_3d Page 4

```

214 C ALONG SOUTH AND NORTH BOUNDARY
215 C
216 DO 33 I=2,NIM
217 IK = (I-1)*NK + K
218 IJK = LK(K) + LI(1) + 2
219 IF(BCFS(IK).NE.6) THEN
220 DUU2(IJK) = 0.5*(-3.0*U2(IJK) + 4.0*U2(IJK+1) - U2(IJK+2))
221 DVU2(IJK) = 0.5*(-3.0*V2(IJK) + 4.0*V2(IJK+1) - V2(IJK+2))
222 DWU2(IJK) = 0.5*(-3.0*W2(IJK) + 4.0*W2(IJK+1) - W2(IJK+2))
223 DUU2(IJK) = 0.5*(-3.0*U2(IJK) + 4.0*U2(IJK+1) - U2(IJK+2))
224 DVU2(IJK) = 0.5*(-3.0*V2(IJK) + 4.0*V2(IJK+1) - V2(IJK+2))
225 DWU2(IJK) = 0.5*(-3.0*W2(IJK) + 4.0*W2(IJK+1) - W2(IJK+2))
226 ENDIF
227 IJK = LK(K) + LI(1) + NJM
228 IF(BCFN(IK).NE.6) THEN
229 DUU2(IJK) = 0.5*(U2(IJK-2) - 4.0*U2(IJK-1) + 3.0*U2(IJK))
230 DVU2(IJK) = 0.5*(V2(IJK-2) - 4.0*V2(IJK-1) + 3.0*V2(IJK))
231 DWU2(IJK) = 0.5*(W2(IJK-2) - 4.0*W2(IJK-1) + 3.0*W2(IJK))
232 DUU2(IJK) = 0.5*(U2(IJK-2) - 4.0*U2(IJK-1) + 3.0*U2(IJK))
233 DVU2(IJK) = 0.5*(V2(IJK-2) - 4.0*V2(IJK-1) + 3.0*V2(IJK))
234 DWU2(IJK) = 0.5*(W2(IJK-2) - 4.0*W2(IJK-1) + 3.0*W2(IJK))
235 ENDIF
236 C
237 C
238 C
239 C ALONG BOTTOM AND TOP BOUNDARY
240 C
241 DO 34 J=2,NJM
242 DO 34 J=2,NJM
243 IJK = LI(1) + LK(2) + J
244 IKJP = IJK + NIJ
245 IJRP = IJK + NIJ
246 IJ = LI(1) + J
247 IF(BCFB(IJ).NE.6) THEN
248 DUU3(IJK) = 0.5*(-3.0*U2(IJK) + 4.0*U2(IJRP) - U2(IJRP))
249 DVU3(IJK) = 0.5*(-3.0*V2(IJK) + 4.0*V2(IJRP) - V2(IJRP))
250 DWU3(IJK) = 0.5*(-3.0*W2(IJK) + 4.0*W2(IJRP) - W2(IJRP))
251 DUU3(IJK) = 0.5*(-3.0*U2(IJK) + 4.0*U2(IJRP) - U2(IJRP))
252 DVU3(IJK) = 0.5*(-3.0*V2(IJK) + 4.0*V2(IJRP) - V2(IJRP))
253 DWU3(IJK) = 0.5*(-3.0*W2(IJK) + 4.0*W2(IJRP) - W2(IJRP))
254 ENDIF
255 IJK = LK(NKM) + LI(1) + J
256 IJEM = IJK - NIJ
257 IJKNM = IJK - NIJ
258 IF(BCFT(IJ).NE.6) THEN
259 DUU3(IJK) = 0.5*(U2(IJKNM) - 4.0*U2(IJEM) + 3.0*U2(IJK))
260 DVU3(IJK) = 0.5*(V2(IJKNM) - 4.0*V2(IJEM) + 3.0*V2(IJK))
261 DWU3(IJK) = 0.5*(W2(IJKNM) - 4.0*W2(IJEM) + 3.0*W2(IJK))
262 DUU3(IJK) = 0.5*(U2(IJKNM) - 4.0*U2(IJEM) + 3.0*U2(IJK))
263 DVU3(IJK) = 0.5*(V2(IJKNM) - 4.0*V2(IJEM) + 3.0*V2(IJK))
264 DWU3(IJK) = 0.5*(W2(IJKNM) - 4.0*W2(IJEM) + 3.0*W2(IJK))
265 ENDIF
266 C
267 C
268 C FOR PERIODICITY ALONG THE SOUTH AND NORTH BOUNDARY
269 C
270 DO 35 I=2,NIM
271 II = (I-1)*NK
272 DO 35 K=2,NKM
273 IK = II + K
274 IF(BCFS(IK).NE.5) GO TO 35
275 IJK = LK(K) + LI(1) + 2
276 IJMK = LK(K) + LI(1) + NJM
277 DUU2(IJK) = 0.5*(U2(IJMK+1) - U2(IJMK))
278 DVU2(IJK) = 0.5*(V2(IJMK+1) - V2(IJMK))
279 DWU2(IJK) = 0.5*(W2(IJMK+1) - W2(IJMK))
280 DUU2(IJK) = 0.5*(U2(IJMK+1) - U2(IJMK))
281 DVU2(IJK) = 0.5*(V2(IJMK+1) - V2(IJMK))
282 DWU2(IJK) = 0.5*(W2(IJMK+1) - W2(IJMK))
283 IJK = LK(K) + LI(1) + NJM
284 IJPK = LK(K) + LI(1) + 2

```

Oct 12 1996 16:37	asmod_3d	Page 5
285 DUU2(IJK) = 0.5*(U2(IJPK) - U2(IJK-1))		
286 DVV2(IJK) = 0.5*(V2(IJPK) - V2(IJK-1))		
287 DWW2(IJK) = 0.5*(W2(IJPK) - W2(IJK-1))		
288 DUU2(IJK) = 0.5*(U2(IJPK) - U2(IJK-1))		
289 DVV2(IJK) = 0.5*(V2(IJPK) - V2(IJK-1))		
290 DWW2(IJK) = 0.5*(W2(IJPK) - W2(IJK-1))		
291 35 CONTINUE		
292 C		
293 DO 50 K=2,NKM		
294 DO 50 I=2,NIM		
295 LK = LI(I) + LK(K)		
296 DO 50 J=2,NJM		
297 C		
298 LJK = LK + J		
299 IFJK = IJK + NJ		
300 IFWK = IWK + NJ		
301 IJPK = IJK + NIJ		
302 IJWK = IWK + NIJ		
303 IJWK=IJK-1		
304 C		
305 DXKS=QTR*(X(IJJK)+X(IJWK)+X(IJRM)+X(IJRM-1)-X(IMJK)-X(IMJK-1))-		
306 X(IJRM-NJ)-X(IJRM-NJ-1))		
307 DXET=QTR*(X(IJJK)+X(IJWK)+X(IMJK)+X(IMJK-1)-X(IJRM-NJ)-		
308 X(IJRM-NJ-1)-X(IMJK-1)-X(IJRM-NJ-1))		
309 DXZD=QTR*(X(IJJK)+X(IJWK)+X(IMJK)+X(IMJK-1)-X(IJRM-NJ)-		
310 X(IJRM-NJ-1)-X(IJRM-NJ-1))		
311 DYKS=QTR*(Y(IJJK)+Y(IJWK)+Y(IJRM)+Y(IJRM-1)-Y(IMJK)-Y(IMJK-1)-		
312 Y(IJRM-NJ)-Y(IJRM-NJ-1))		
313 DYET=QTR*(Y(IJJK)+Y(IJWK)+Y(IMJK)+Y(IMJK-1)-Y(IJRM-NJ)-		
314 Y(IJRM-NJ-1)-Y(IMJK-1)-Y(IJRM-NJ-1))		
315 DYZD=QTR*(Y(IJJK)+Y(IMJK)+Y(IMJK)+Y(IMJK-1)-Y(IJRM-NJ)-		
316 Y(IJRM-NJ-1)-Y(IJRM-NJ-1))		
317 DZKS=QTR*(Z(IJJK)+Z(IJWK)+Z(IJRM)+Z(IJRM-1)-Z(IMJK)-Z(IMJK-1)-		
318 Z(IJRM-NJ)-Z(IJRM-NJ-1))		
319 DZET=QTR*(Z(IJJK)+Z(IJWK)+Z(IMJK)+Z(IMJK-1)-Z(IJRM-NJ)-		
320 Z(IJRM-NJ-1)-Z(IJRM-NJ-1))		
321 DZDZ=QTR*(Z(IJJK)+Z(IJWK)+Z(IMJK)+Z(IMJK-1)-Z(IJRM-NJ)-		
322 Z(IJRM-NJ)-Z(IJRM-NJ-1))		
323 C..... INLINE		
324 B11=DYET*DZDZ-DYDZ*DZET		
325 B12=DXZD*DZET-DXET*DZDZ		
326 B13=DXET*DYDZ-DYET*DXZD		
327 B21=DXKS*DYDZ-DYDZ*DXKS		
328 B22=DXKS*DZDZ-DZDZ*DXKS		
329 B23=DXZD*DYKS-DYKS*DXZD		
330 B31=DYKS*DZET-DZET*DYKS		
331 B32=DZKS*DXET-DXET*DZKS		
332 B33=DXKS*DYET-DXET*DYKS		
333 C Calculate stress gradient source terms		
334 SUASM(IJK)=B11*DUU1(IJK)-B21*DUU2(IJK)-B31*DUU3(IJK)		
335 & -B12*DUU1(IJK)-B22*DUU2(IJK)-B32*DUU3(IJK)		
336 & -B13*DUU1(IJK)-B23*DUU2(IJK)-B33*DUU3(IJK)		
337 & SVASM(IJK)=-B11*DUV1(IJK)-B21*DUV2(IJK)-B31*DUV3(IJK)		
338 & -B12*DUV1(IJK)-B22*DUV2(IJK)-B32*DUV3(IJK)		
339 & -B13*DUV1(IJK)-B23*DUV2(IJK)-B33*DUV3(IJK)		
340 & SWASM(IJK)=-B11*DWU1(IJK)-B21*DWU2(IJK)-B31*DWU3(IJK)		
341 & -B12*DWU1(IJK)-B22*DWU2(IJK)-B32*DWU3(IJK)		
342 & -B13*DWU1(IJK)-B23*DWU2(IJK)-B33*DWU3(IJK)		
343 & CONTINUE		
344 50		
345 C		
346 RETURN		
347 END		
348 C		
349 C*****		
350 SUBROUTINE CALUIU3(NIM,NJM,NKM,LK,LI,U2,V2,W2,UV,UW,VW,		
351 & GENTB,GENTT,GENTN,GENTW,GENTE, GEN,		
352 & BCFW,BCFE,BCFS,BCFN,BCFB,BCFT,		
353 & X,Y,Z,FX,FY,FZ,U,V,W,VOL,		
354 & TE,ED,VIS,OMX,OMY,OMZ,DENSIT,VISCOS)		
355 C*****		

```

356 C
357 INCLUDE 'param.h'
358 INCLUDE 'asmod.h'
359 C
360 DIMENSION LK(JZ),LI(IJK),
361 & X(JXYZ),Y(JXYZ),Z(JXYZ),VOL(JXYZ),
362 & FX(JXYZ),FY(JXYZ),FZ(JXYZ),
363 DIMENSION U(JXYZ),V(JXYZ),W(JXYZ),
364 & TE(JXYZ),ED(JXYZ),VIS(JXYZ)
365 DIMENSION U2(JXYZ),V2(JXYZ),W2(JXYZ),
366 & UV(JXYZ),UW(JXYZ),VW(JXYZ)
367 DIMENSION GEN(JXYZ)
368 DIMENSION GENTB(JXY),GENTT(JXY),GENTS(JXZ),GENTN(JXZ),
369 & GENTW(JYZ),GENTE(JYZ)
370 DIMENSION BCFW(JYZ),BCFE(JYZ),BCFS(JXZ),BCFN(JXZ),
371 & BCFB(JXY),BCFT(JXY)
372 C
373 INTEGER BCFW,BCFE,BCFS,BCFN,BCFB,BCFT
374 DIMENSION A(6,6),B(6)
375 C
376 C---CALCULATE ALGEBRAIC STRESS EQUATIONS IN THE FORM
377 C
378 C- A11*U2(IJK) + A12*V2(IJK) + A13*W2(IJK) +
379 C- A14*UV(IJK) + A15*VW(IJK) + A16*(UW(IJK) = B1
380 C
381 DO 10 K=2,NKM
382 LKK=LK(K)
383 DO 10 I=2,NIM
384 LII=LI(I)
385 DO 10 J=2,NJM
386 IG=LII+J
387 IJK=LKK+IJ
388 C
389 EDK=ED(IJK)/(TE(IJK)*SMALL)
390 AUX=(1.-CZASM)/(CLASM*ED(IJK)+GEN(IJK)-ED(IJK))
391 AUX=1./(AUX*TE(IJK)+SMALL)
392 C--U2-EQUATION
393 A(1,1)=1.5*AUX+2.*DUDX(IJK)
394 A(1,2)=-DUDY(IJK)
395 A(1,3)=-DWDZ(IJK)
396 A(1,4)=2.*DUDY(IJK)-DUDX(IJK)-6.*COMEGA*OMZ
397 A(1,5)=-DVDZ(IJK)-DWDY(IJK)
398 A(1,6)=2.*DUDZ(IJK)-DWDX(IJK)+6.*COMEGA*OMY
399 C--V2-EQUATION
400 A(2,1)=-DUDX(IJK)
401 A(2,2)=1.5*AUX+2.*DUDY(IJK)
402 A(2,3)=-DWDZ(IJK)
403 A(2,4)=2.*DUDZ(IJK)-DWDY(IJK)-6.*COMEGA*OMZ
404 A(2,5)=2.*DUDZ(IJK)-DWDX(IJK)-6.*COMEGA*OMX
405 A(2,6)=-DUDZ(IJK)-DWDX(IJK)
406 C--W2-EQUATION
407 A(3,1)=-DUDX(IJK)
408 A(3,2)=-DUDY(IJK)
409 A(3,3)=1.5*AUX+2.*DWDZ(IJK)
410 A(3,4)=-DUDY(IJK)-DWDX(IJK)
411 A(3,5)=-DVDZ(IJK)+2.*DWDY(IJK)+6.*COMEGA*OMX
412 A(3,6)=-DUDZ(IJK)+2.*DWDX(IJK)-6.*COMEGA*OMY
413 C--UV-EQUATION
414 A(4,1)=DUDX(IJK)+2.*COMEGA*OMZ
415 A(4,2)=DUDY(IJK)-2.*COMEGA*OMZ
416 A(4,3)=0.0
417 A(4,4)=AUX+DUDX(IJK)+DUDY(IJK)
418 A(4,5)=DUDZ(IJK)+2.*COMEGA*OMY
419 A(4,6)=DWDZ(IJK)-2.*COMEGA*OMX
420 C--VW-EQUATION
421 A(5,1)=0.0
422 A(5,2)=DWDY(IJK)+2.*COMEGA*OMX
423 A(5,3)=DWDZ(IJK)-2.*COMEGA*OMX
424 A(5,4)=DWDX(IJK)-2.*COMEGA*OMY
425 A(5,5)=AUX+DUDY(IJK)+DWDZ(IJK)
426 A(5,6)=DWDX(IJK)+2.*COMEGA*OMZ

```

Oct 12 1996 16:37	asmod_3d	Page 7
427 C--UW-EQUATION		
428 A(6,1)=DWDZ(IJK)-2.*OMEGA*OMY		
429 A(6,2)=0.		
430 A(6,3)=DUDZ(IJK)+2.*OMEGA*OMY		
431 A(6,4)=DWDY(IJK)+2.*OMEGA*OMX		
432 A(6,5)=DUDY(IJK)-2.*OMEGA*OMZ		
433 A(6,6)=AUX+DUDX(IJK)+DWDZ(IJK)		
434 C		
435 C--RIGHT HAND SIDE OF EQUATION		
436 B(1)=AUX*TE(IJK)		
437 B(2)=AUX*TE(IJK)		
438 B(3)=AUX*TE(IJK)		
439 B(4)=0.		
440 B(5)=0.		
441 B(6)=0.		
442 IF(WREFON.EQ.1) THEN		
443 B1=1./ (1.-C2ASM)		
444 B2=C1P*ED(IJK)/(TE(IJK)*SMALL)		
445 B3=C2ASM*C2P		
446 C--RHS OF U2-EQUATION		
447 FW1=B2*(-2.*U2(IJK)*FUNX(IJK)+V2(IJK)*FUNY(IJK)+		
448 & W2(IJK)*FUNZ(IJK)-		
449 & UV(IJK)*FUNX(IJK)-UV(IJK)*FUNXZ(IJK)+2.*VW(IJK)*FUNYZ(IJK))		
450 FW2=B3*(2.*(P11(IJK)-2./3.*GEN(IJK))*FUNX(IJK)-		
451 & (P22(IJK)-2./3.*GEN(IJK))*FUNY(IJK)-		
452 & (P33(IJK)-2./3.*GEN(IJK))*FUNZ(IJK)+		
453 & P12(IJK)*FUNX(IJK)+P13(IJK)*FUNXZ(IJK)-		
454 & 2.*P23(IJK)*FUNYZ(IJK))		
455 B(1)=B(1)+1.5*B1*(FW1+FW2)		
456 C--RHS OF V2-EQUATION		
457 FW1=B2*(U2(IJK)*FUNX(IJK)-2.*V2(IJK)*FUNY(IJK)+		
458 & W2(IJK)*FUNZ(IJK)-		
459 & UV(IJK)*FUNX(IJK)+2.*UV(IJK)*FUNXZ(IJK)-VW(IJK)*FUNYZ(IJK))		
460 FW2=B3*(-(P11(IJK)-2./3.*GEN(IJK))*FUNX(IJK)+P12(IJK)*FUNXZ(IJK)-		
461 & 2.*P13(IJK)*FUNXZ(IJK)+2.*(P22(IJK)-2./3.*GEN(IJK))*FUNY(IJK)+		
462 & P23(IJK)*FUNYZ(IJK)-(P33(IJK)-2./3.*GEN(IJK))*FUNZ(IJK))		
463 B(2)=B(2)+1.5*B1*(FW1+FW2)		
464 C--RHS OF W2-EQUATION		
465 FW1=B2*(U2(IJK)*FUNX(IJK)+V2(IJK)*FUNY(IJK)-		
466 & 2.*W2(IJK)*FUNZ(IJK)+		
467 & 2.*UV(IJK)*FUNX(IJK)-UV(IJK)*FUNXZ(IJK)-VW(IJK)*FUNYZ(IJK))		
468 FW2=B3*(-(P11(IJK)-2./3.*GEN(IJK))*FUNX(IJK)+P12(IJK)*FUNXZ(IJK)-		
469 & 2.*P13(IJK)*FUNXZ(IJK)+		
470 & P13(IJK)*FUNXZ(IJK)-(P22(IJK)-2./3.*GEN(IJK))*FUNY(IJK)+		
471 & P33(IJK)*FUNYZ(IJK)+2.*(P33(IJK)-2./3.*GEN(IJK))*FUNZ(IJK))		
472 B(3)=B(3)+1.5*(FW1+FW2)		
473 C--RHS OF UV-EQUATION		
474 FW1=-1.5*B2*(UV(IJK)*FUNX(IJK)+FUNY(IJK))*U2(IJK)+		
475 & V2(IJK)*FUNX(IJK)+		
476 & UV(IJK)*FUNZ(IJK)+VW(IJK)*FUNXZ(IJK))		
477 FW2=-1.5*B3*((P11(IJK)-2./3.*GEN(IJK))*FUNX(IJK)+		
478 & P12(IJK)*FUNXZ(IJK)+FUNY(IJK)+		
479 & (P22(IJK)-2./3.*GEN(IJK))*FUNX(IJK)+P13(IJK)*FUNYZ(IJK)+		
480 & P23(IJK)*FUNXZ(IJK))		
481 B(4)=B(4)+B1*(FW1+FW2)		
482 C--RHS OF VW-EQUATION		
483 FW1=-1.5*B2*(UV(IJK)*FUNX(IJK)+FUNY(IJK)+UV(IJK)*FUNXZ(IJK)+		
484 & VW(IJK)*FUNXZ(IJK)+FUNY(IJK)+VW(IJK)*FUNYZ(IJK)+		
485 & W2(IJK)*FUNZ(IJK))		
486 FW2=-1.5*B3*(P12(IJK)*FUNXZ(IJK)+P13(IJK)*FUNXZ(IJK)+		
487 & (P22(IJK)-2./3.*GEN(IJK))*FUNYZ(IJK)+		
488 & P23(IJK)*FUNY(IJK)+FUNZ(IJK)+		
489 & (P33(IJK)-2./3.*GEN(IJK))*FUNY(IJK)+FUNZ(IJK))		
490 B(5)=B(5)+B1*(FW1+FW2)		
491 C--RHS OF UW-EQUATION		
492 FW1=-1.5*B2*(UV(IJK)*FUNX(IJK)+FUNY(IJK))*U2(IJK)*FUNXZ(IJK)+		
493 & W2(IJK)*FUNXZ(IJK)+UV(IJK)*FUNYZ(IJK)+VW(IJK)*FUNXZ(IJK)+		
494 & FW2=-1.5*B3*((P11(IJK)-2./3.*GEN(IJK))*FUNXZ(IJK)+		
495 & P13(IJK)*FUNXZ(IJK)+FUNZ(IJK)+P12(IJK)*FUNYZ(IJK)+		
496 & P23(IJK)*FUNXZ(IJK)+P33(IJK)-2./3.*GEN(IJK))*FUNXZ(IJK)+		
497 B(6)=B(6)+B1*(FW1+FW2)		

Oct 12 1996 16:37	asmod_3d	Page 8
498 C		
499 ENDIF		
500 C		
501 C-----SOLVE THE 6X6 ALGEBRAIC EQUATIONS		
502 CALL SOLV(A,B,6)		
503 C		
504 C---N.B. SOLUTION IS IN B(1...6) FOR U2,V2,...,UW		
505 C		
506 U2(IJK)=B(1)*RELT*(1.-RELT)*U2(IJK)		
507 V2(IJK)=B(2)*RELT*(1.-RELT)*V2(IJK)		
508 W2(IJK)=B(3)*RELT*(1.-RELT)*W2(IJK)		
509 UV(IJK)=B(4)*RELT*(1.-RELT)*UV(IJK)		
510 VW(IJK)=B(5)*RELT*(1.-RELT)*VW(IJK)		
511 UW(IJK)=B(6)*RELT*(1.-RELT)*UW(IJK)		
512 C		
513 C		
514 C--LIMIT NORMAL STRESSES		
515 C		
516 TAUMAX=2.*TE(IJK)		
517 TAUMIN=0.		
518 U2(IJK)=AMIN(U2(IJK),TAUMAX)		
519 V2(IJK)=AMIN(V2(IJK),TAUMIN)		
520 W2(IJK)=AMIN(W2(IJK),TAUMAX)		
521 V2(IJK)=AMIN(V2(IJK),TAUMIN)		
522 W2(IJK)=AMIN(W2(IJK),TAUMAX)		
523 W2(IJK)=AMIN(W2(IJK),TAUMIN)		
524 C		
525 10		
526 C		
527 CALL WALSTRS(NIM,NJM,NKM,LI,LK,FX,FY,FZ,		
528 & X,Y,Z,VOL,U,V,W,VIS,VISCOS,DENSIT,TE,		
529 & BCFW,BCFE,BCFS,BCFN,BCFB,BCFT,		
530 & U2,V2,W2,UV,VW,		
531 &)		
532 C		
533 C--RECALCULATE THE PRODUCTION TERMS		
534 C		
535 C		
536 CALL CALPIJ(NIM,NJM,NKM,LI,LK,FX,FY,FZ,		
537 & X,Y,Z,VOL,U,V,W,VIS,VISCOS,DENSIT,TE,		
538 & GENTB,GENTT,GENTS,GENTW,GENTE,		
539 & BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)		
540 C		
541 RETURN		
542 END		
543 C		
544 C*****		
545 SUBROUTINE WALSTRS(NIM,NJM,NKM,LI,LK,FX,FY,FZ,		
546 & X,Y,Z,VOL,U,V,W,VIS,VISCOS,DENSIT,TE,		
547 & BCFW,BCFE,BCFS,BCFN,BCFB,BCFT,		
548 & U2,V2,W2,UV,VW)		
549 C*****		
550 C		
551 INCLUDE 'param.h'		
552 INCLUDE 'asmod.h'		
553 C		
554 DIMENSION LK(JZ),LI(JX),		
555 & X(JXYZ),Y(JXYZ),Z(JXYZ),VOL(JXYZ),		
556 & FX(JXYZ),FY(JXYZ),FZ(JXYZ)		
557 DIMENSION U(JXYZ),V(JXYZ),W(JXYZ),		
558 & TE(JXYZ),VIS(JXYZ)		
559 DIMENSION U2(JXYZ),V2(JXYZ),W2(JXYZ),		
560 & UV(JXYZ),UW(JXYZ),VW(JXYZ)		
561 DIMENSION BCFW(JYZ),BCFE(JYZ),BCFS(JYZ),		
562 & BCFB(JXY),BCFT(JXY)		
563 INTEGER BCFW,BCFE,BCFS,BCFN,BCFB,BCFT		
564 C		
565 C----- CALCULATE MEAN VELOCITY GRADIENTS		
566 C		
567 CALL PHGRAD(U,DUDX,DUDY,DUDZ,		
568 & NIM,NJM,NKM,LI,LK,FX,FY,FZ,		

Oct 12 1996 16:37	asmod_3d	Page 9
569	& X,Y,Z,VOL,	
570	& BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)	
571	CALL PHGRAD(V,DVDZ,DVDY,DVDZ,	
572	& NIM,NJM,NKM,LI,FK,FX,FY,FZ,	
573	& X,Y,Z,VOL,	
574	& BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)	
575	CALL PHGRAD(W,DWDZ,DWDY,DWDZ,	
576	& NIM,NJM,NKM,LI,FK,FX,FY,FZ,	
577	& X,Y,Z,VOL,	
578	& BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)	
579	C	
580	C---SET STRESS COMPONENTS AT THE BOUNDARY EDGES.	
581	C	
582	C---WEST & EAST FACE	
583	DO K=2,NKM	
584	KK=(K-1)*NJ	
585	DO J=2,NJM	
586	JK=KK+J	
587	IF(BCFN(JK).EQ.4) THEN	
588	IJK=LK(K)+LI(I)+J	
589	TAUMAX=2.*TE(IJK)	
590	TAUMIN=0.0	
591	VIST=VIS(IJK)-VISCOS	
592	U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)	
593	V2(IJK)=-2.*VIST*DUDY(IJK)+2./3.*DENSIT*TE(IJK)	
594	W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)	
595	UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))	
596	VM(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))	
597	UM(IJK)=-VIST*(DVDZ(IJK)+DWDZ(IJK))	
598	U2(IJK)=AMINI(U2(IJK),TAUMAX)	
599	U2(IJK)=AMAXI(U2(IJK),TAUMIN)	
600	V2(IJK)=AMINI(V2(IJK),TAUMAX)	
601	V2(IJK)=AMAXI(V2(IJK),TAUMIN)	
602	W2(IJK)=AMINI(W2(IJK),TAUMAX)	
603	W2(IJK)=AMAXI(W2(IJK),TAUMIN)	
604	ENDIF	
605	IF(BCFE(JK).EQ.4) THEN	
606	IJK=LK(K)+LI(NIM)+J	
607	TAUMAX=2.*TE(IJK)	
608	TAUMIN=0.0	
609	VIST=VIS(IJK)-VISCOS	
610	U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)	
611	V2(IJK)=-2.*VIST*DUDY(IJK)+2./3.*DENSIT*TE(IJK)	
612	W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)	
613	UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))	
614	VM(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))	
615	UM(IJK)=-VIST*(DVDZ(IJK)+DWDZ(IJK))	
616	U2(IJK)=AMINI(U2(IJK),TAUMAX)	
617	U2(IJK)=AMAXI(U2(IJK),TAUMIN)	
618	V2(IJK)=AMINI(V2(IJK),TAUMAX)	
619	V2(IJK)=AMAXI(V2(IJK),TAUMIN)	
620	W2(IJK)=AMINI(W2(IJK),TAUMAX)	
621	W2(IJK)=AMAXI(W2(IJK),TAUMIN)	
622	ENDIF	
623	ENDDO	
624	ENDDO	
625	C	
626	C---SOUTH & NORTH FACE	
627	DO I=2,NIM	
628	II=(I-1)*NK	
629	DO K=2,NKM	
630	IK=II+K	
631	IF(BCFS(IK).EQ.4) THEN	
632	IJK=LK(K)+LI(I)+2	
633	TAUMAX=2.*TE(IJK)	
634	TAUMIN=0.0	
635	VIST=VIS(IJK)-VISCOS	
636	U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)	
637	V2(IJK)=-2.*VIST*DUDY(IJK)+2./3.*DENSIT*TE(IJK)	
638	W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)	
639	UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))	

Oct 12 1996 16:37	asmod_3d	Page 10
640	VM(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))	
641	UM(IJK)=-VIST*(DVDZ(IJK)+DWDZ(IJK))	
642	U2(IJK)=AMINI(U2(IJK),TAUMAX)	
643	U2(IJK)=AMAXI(U2(IJK),TAUMIN)	
644	V2(IJK)=AMINI(V2(IJK),TAUMAX)	
645	V2(IJK)=AMAXI(V2(IJK),TAUMIN)	
646	W2(IJK)=AMINI(W2(IJK),TAUMAX)	
647	W2(IJK)=AMAXI(W2(IJK),TAUMIN)	
648	ENDIF	
649	IF(BCFN(IK).EQ.4) THEN	
650	IJK=LK(K)+LI(I)+NJM	
651	TAUMAX=2.*TE(IJK)	
652	TAUMIN=0.0	
653	VIST=VIS(IJK)-VISCOS	
654	U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)	
655	V2(IJK)=-2.*VIST*DUDY(IJK)+2./3.*DENSIT*TE(IJK)	
656	W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)	
657	UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))	
658	VM(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))	
659	UM(IJK)=-VIST*(DVDZ(IJK)+DWDZ(IJK))	
660	U2(IJK)=AMINI(U2(IJK),TAUMAX)	
661	U2(IJK)=AMAXI(U2(IJK),TAUMIN)	
662	V2(IJK)=AMINI(V2(IJK),TAUMAX)	
663	V2(IJK)=AMAXI(V2(IJK),TAUMIN)	
664	W2(IJK)=AMINI(W2(IJK),TAUMAX)	
665	W2(IJK)=AMAXI(W2(IJK),TAUMIN)	
666	ENDIF	
667	ENDDO	
668	ENDDO	
669	C---BOTTOM & TOP FACE	
670	DO I=2,NIM	
671	II=(I-1)*NJ	
672	DO J=2,NJM	
673	IJ=LI(I)+J	
674	IF(BCFB(IJ).EQ.4) THEN	
675	IJK=LK(2)+LI(I)+J	
676	TAUMAX=2.*TE(IJK)	
677	TAUMIN=0.0	
678	VIST=VIS(IJK)-VISCOS	
679	U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)	
680	V2(IJK)=-2.*VIST*DUDY(IJK)+2./3.*DENSIT*TE(IJK)	
681	W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)	
682	UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))	
683	VM(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))	
684	UM(IJK)=-VIST*(DVDZ(IJK)+DWDZ(IJK))	
685	U2(IJK)=AMINI(U2(IJK),TAUMAX)	
686	U2(IJK)=AMAXI(U2(IJK),TAUMIN)	
687	V2(IJK)=AMINI(V2(IJK),TAUMAX)	
688	V2(IJK)=AMAXI(V2(IJK),TAUMIN)	
689	W2(IJK)=AMINI(W2(IJK),TAUMAX)	
690	W2(IJK)=AMAXI(W2(IJK),TAUMIN)	
691	ENDIF	
692	IF(BCFT(IJ).EQ.4) THEN	
693	IJK=LK(NKM)+LI(I)+J	
694	TAUMAX=2.*TE(IJK)	
695	TAUMIN=0.0	
696	VIST=VIS(IJK)-VISCOS	
697	U2(IJK)=-2.*VIST*DUDX(IJK)+2./3.*DENSIT*TE(IJK)	
698	V2(IJK)=-2.*VIST*DUDY(IJK)+2./3.*DENSIT*TE(IJK)	
699	W2(IJK)=-2.*VIST*DWDZ(IJK)+2./3.*DENSIT*TE(IJK)	
700	UV(IJK)=-VIST*(DUDY(IJK)+DVDX(IJK))	
701	VM(IJK)=-VIST*(DVDZ(IJK)+DWDY(IJK))	
702	UM(IJK)=-VIST*(DVDZ(IJK)+DWDZ(IJK))	
703	U2(IJK)=AMINI(U2(IJK),TAUMAX)	
704	U2(IJK)=AMAXI(U2(IJK),TAUMIN)	
705	V2(IJK)=AMINI(V2(IJK),TAUMAX)	
706	V2(IJK)=AMAXI(V2(IJK),TAUMIN)	
707	W2(IJK)=AMINI(W2(IJK),TAUMAX)	
708	W2(IJK)=AMAXI(W2(IJK),TAUMIN)	
709	ENDIF	
710	ENDDO	

Oct 12 1996 16:37	asmod_3d	Page 11
711 C	ENDDO	
712 C	RETURN	
713	END	
714		
715 C		
716 C		
717	SUBROUTINE SOLV(A,BB,N)	
718 C		
719 C	A GAUSS ELIMINATION SOLVER	
720 C		
721 C	DIMENSION A(N,N),B(10),C(10),BB(N),X(10),MWE(10)	
722 C		
723	EP =1.E-19	
724	DO 10 J=1,N	
725 10	MWE(J)=J	
726	DO 20 I=1,N	
727	Y=0.	
728	DO 30 J=I,N	
729	IF (ABS(A(I,J)).LT.ABS(Y)) GOTO 30	
730	K=J	
731	Y=A(I,J)	
732 30	CONTINUE	
733 C		
734	IF (ABS(Y).LT.EP) THEN	
735	WRITE(*,*)	
736	WRITE(9,*)	
737	DO 35 IA=1,N	
738	WRITE(*,1000) (A(IA,JA),JA=1,N)	
739 35	CONTINUE	
740	PRINT*, 'THERE IS NO CONVERSE MATRIX'	
741	STOP 2222	
742	ENDIF	
743 C		
744	Y=1./Y	
745	DO 40 J=1,N	
746	C(J)=A(J,K)	
747	C(J,K)=A(J,I)	
748	A(J,I)=-C(J)*Y	
749	B(J)=A(I,J)*Y	
750 40	A(I,J)=A(I,J)*Y	
751	A(I,I)=Y	
752	J=MWE(I)	
753	MWE(I)=MWE(K)	
754	MWE(K)=J	
755	DO 11 K=1,N	
756	IF (K.EQ.I) GOTO 11	
757	DO 12 J=1,N	
758	IF (J.EQ.I) GOTO 12	
759	A(K,J)=A(K,J)-B(J)*C(K)	
760 12	CONTINUE	
761 11	CONTINUE	
762 20	CONTINUE	
763	DO 33 I=1,N	
764	DO 44 K=1,N	
765	IF (MWE(K).EQ.I) GOTO 55	
766 44	CONTINUE	
767 55	IF (K.EQ.I) GOTO 33	
768	DO 66 J=1,N	
769	W=A(I,J)	
770	A(I,J)=A(K,J)	
771 66	A(K,J)=W	
772	I=MWE(I)	
773	MWE(I)=MWE(K)	
774	MWE(K)=I	
775 33	CONTINUE	
776 1000	FORMAT(4X,IP5E13.4)	
777	DO 50 I=1,N	
778	X(I)=0.	
779	DO 50 J=1,N	
780 50	X(I)=X(I)+A(I,J)*BB(J)	
781	DO 60 I=1,N	

Oct 12 1996 16:37	asmod_3d	Page 12
782 60	BB(I)=X(I)	
783	RETURN	
784	END	
785 C		
786 C		
787	SUBROUTINE CALPIJ(NIM,NM,NKM,LI,LK,FX,FY,FZ,	
788	& X,Y,Z,VOL, U,V,W, U2,V2,W2, UV,UW,VW, GEN,	
789	& GENTB,GENTT,GENTS,GENTN,GENTW,GENTE,	
790	& BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)	
791 C		
792 C		
793	INCLUDE 'param.h'	
794	INCLUDE 'asmod.h'	
795 C		
796	DIMENSION LK(JZ),LI(JX),	
797	& X(JXYZ),Y(JXYZ),Z(JXYZ),VOL(JXYZ),	
798	& FX(JXYZ),FY(JXYZ),FZ(JXYZ)	
799	DIMENSION U(JXYZ),V(JXYZ),W(JXYZ)	
800	DIMENSION U2(JXYZ),V2(JXYZ),W2(JXYZ),	
801	& UV(JXYZ),UW(JXYZ),VW(JXYZ)	
802	DIMENSION GEN(JXYZ)	
803	DIMENSION BCFW(JYZ),BCFE(JYZ),BCFS(JYZ),BCFN(JXZ),	
804	& BCFB(JXY),BCFT(JXY)	
805	DIMENSION GENTB(JXY),GENTT(JXY),GENTS(JXZ),GENTN(JXZ),	
806	& GENTW(JYZ),GENTE(JYZ)	
807	INTEGER BCFW,BCFE,BCFS,BCFN,BCFB,BCFT	
808 C		
809 C	----- CALCULATE MEAN VELOCITY GRADIENTS	
810 C		
811	CALL PHGRAD(U,DUDX,DUDY,DUDZ,	
812	& NIM,NJM,NKM,LI,LK,FX,FY,FZ,	
813	& X,Y,Z,VOL,	
814	& BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)	
815	CALL PHGRAD(V,DVDX,DVDY,DVDZ,	
816	& NIM,NJM,NKM,LI,LK,FX,FY,FZ,	
817	& X,Y,Z,VOL,	
818	& BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)	
819	CALL PHGRAD(W,DWDX,DWDY,DWDZ,	
820	& NIM,NJM,NKM,LI,LK,FX,FY,FZ,	
821	& X,Y,Z,VOL,	
822	& BCFW,BCFE,BCFS,BCFN,BCFB,BCFT)	
823 C		
824	DO K=2,NKM	
825	DO I=2,NIM	
826	DO J=2,NJM	
827	IJK=LK(K)+LI(I)+J	
828 C		
829	P11(IJK)=-2.*(U2(IJK)*DUDX(IJK)+UV(IJK)*DUDY(IJK)+	
830	& UW(IJK)*DUDZ(IJK))	
831	P22(IJK)=-2.*(UV(IJK)*DVDX(IJK)+V2(IJK)*DVDY(IJK)+	
832	& VW(IJK)*DVDZ(IJK))	
833	P33(IJK)=-2.*(UW(IJK)*DWDX(IJK)+VW(IJK)*DWDY(IJK)+	
834	& W2(IJK)*DWDZ(IJK))	
835	P12(IJK)=- (U2(IJK)*DVDX(IJK)+V2(IJK)*DUDY(IJK)+	
836	& UV(IJK)*DWDZ(IJK) +	
837	& UW(IJK)*DVDZ(IJK)+VW(IJK)*DUDZ(IJK))	
838	P13(IJK)=- (U2(IJK)*DWDX(IJK)+W2(IJK)*DUDZ(IJK)+	
839	& UV(IJK)*DWDY(IJK) +	
840	& UW(IJK)*DVDY(IJK)+VW(IJK)*DUDY(IJK))	
841	P23(IJK)=- (V2(IJK)*DWDY(IJK)+W2(IJK)*DVDZ(IJK)+	
842	& UV(IJK)*DWDX(IJK) +	
843	& UW(IJK)*DWDZ(IJK)+VW(IJK)*DUDX(IJK))	
844 C		
845	GEN(IJK)=HAF*ABS(P11(IJK)+P22(IJK)+P33(IJK))	
846 C		
847	ENDDO	
848	ENDDO	
849	ENDDO	
850 C		
851 C	----- MODIFY GENERATION TERM CLOSE TO A WALL	
852 C		

Oct 12 1996 16:37	asmod_3d	Page 15
995 C		
996 10	CONTINUE	
997 C		
998	RETURN	
999	END	
1000 C		
1001 C	-----	
1002 C--	param.h	
1003 C		
1004	PARAMETER (JX=102)	
1005	PARAMETER (JY=45)	
1006	PARAMETER (JZ=26)	
1007	PARAMETER (JXY=JX*JY)	
1008	PARAMETER (JXZ=JX*JZ)	
1009	PARAMETER (JYZ=JY*JZ)	
1010	PARAMETER (JXYZ=JX*JY*JZ)	
1011 C		
1012	PARAMETER (GREAT=1.0E+30)	
1013	PARAMETER (SMALL=1.0E-30)	
1014	PARAMETER (HAF=0.5)	
1015	PARAMETER (QTR=0.25)	
1016	PARAMETER (AHT=0.125)	
1017	PARAMETER (IEQ=2)	
1018		
1019 C		
1020 C	-----	
1021 C--	asmod.h	
1022 C		
1023 C		
1024 C	COMMON BLOCKS FOR VARIABLES INSIDE 3D ASM MODULE	
1025 C		
1026	COMMON/ASMCB0/	
1027	& NI,NJ,NK,NIJ,NIK,NJK,NIJK	
1028	COMMON/ASMCB1/	
1029	& P11(JXYZ),P22(JXYZ),P33(JXYZ),P12(JXYZ),P13(JXYZ),	
1030	& P23(JXYZ)	
1031	COMMON/ASMCB2/	
1032	& DUDX(JXYZ),DUDY(JXYZ),DUDZ(JXYZ),DVDX(JXYZ),DVDY(JXYZ),	
1033	& DVDZ(JXYZ),DWDX(JXYZ),DWDY(JXYZ),DWDZ(JXYZ)	
1034	COMMON/ASMCB3/	
1035	& FUNX(JXYZ),FUNY(JXYZ),FUNZ(JXYZ),FUNXY(JXYZ),	
1036	& FUNYZ(JXYZ),FUNKZ(JXYZ)	
1037	COMMON/ASMCB4/	
1038	& CLASH,C2ASM,C1P,C2P,WREFON,CONEGA,RELT	
1039 C		

CHAPTER 8

Related Publications and Presentations

During the course of this work, some related turbulence modeling work was published or presented at different meetings. Copies of these papers are listed in this chapter, they include;

- (1) A. H. Hadid and M. M. Sindir "Comparative study of advanced turbulence models for turbomachinery" NACA CP-3174, 1992.

This paper was presented at the advanced Earth-to-Orbit propulsion technology conference held at NASA Marshall Space Flight Center in Huntsville, Alabama on May 19-21, 1992. The work tests different correction to the standard $k-\epsilon$ turbulence models that accounts for streamline curvature and rotations using different near wall treatments.

- (2) A. H. Hadid, M. E. DeCroix and M. M. Sindir "Advanced turbulence models for turbomachinery" NASA CP-3221, 1993.

This paper was presented at the eleventh workshop for computational fluid dynamics applications in rocket propulsion held at NASA Marshall Space Flight Center in Huntsville, Alabama on April 20-22, 1993. The paper outlined the progress of the 2D/axisymmetric single and multi-scale $k-\epsilon$ turbulence module deck developments.

- (3) A. Hadid, M. Sindir, C. Chen and H. Wei "Computations of confined swirling flows with high order turbulence models in a modular form"

This paper was presented at the twelfth workshop for computational fluid dynamics applications in rocket propulsion held at NASA Marshall Space Flight Center in Huntsville, Alabama April-May 1994. The paper presented the status of the 2D/axisymmetric second order closure models using the algebraic and the full Reynolds stress models.

- (4) A. H. Hadid, M. M. Sindir and R. I. Issa "A numerical study of two-dimensional vortex shedding from rectangular cylinders" published in the CFD Journal July, 1992.

This paper presents a test for an anisotropic k - ε turbulence model. This model is an improvement on the standard k - ε model since it can predict Reynolds stress anisotropies without the need to solve additional equations for the stresses.

- (5) A. H. Hadid, N. N. Mansour and O. Zeman "Single point modeling of rotating turbulent flows" Proceedings of the 1994 summer program, CTR, NASA Ames/Stanford University.

This paper tests a new one-point closure model that incorporates the effects of rotation on the power-law decay exponent of the turbulent kinetic energy. A modification to the ε -equation proposed by Zeman using large eddy simulation results was used. A new definition of the mean rotation was proposed based on critical point theory to generalize the effects of rotation on turbulence to arbitrary mean deformations.

COMPARATIVE STUDY OF ADVANCED TURBULENCE MODELS FOR TURBOMACHINERY

A. H. Hadid and M. M. Sindir
CFD Technology Center
Rocketdyne Division, Rockwell International
Mail Code IB39, 6633 Canoga Avenue
Canoga Park, CA 91303

ABSTRACT

Development and assessment of the standard $k-\epsilon$ turbulence model for rotating flows with different near-wall treatments is presented. These include the standard wall function⁽¹⁾, Patel's two-layer model⁽²⁾, and Lam and Bremhorst⁽³⁾ low-Reynolds number model. Two test cases were chosen to validate these models for rotating flows. The first, from Daily and Nece⁽⁴⁾, is for a rotating disk cavity in which recirculation and secondary flows are induced by the rotating element. The second case is that of a confined double concentric jets with a sudden expansion by Roback and Johnson⁽⁵⁾.

It is shown that near-wall effects are important close to rotating walls and that the two-layer model behaves better than the other two near-wall models. For confined swirling flows with fixed walls, the near wall effects are of secondary importance to the Reynold's stress anisotropy.

INTRODUCTION

Accurate predictions of turbulent flows are crucial to the design and analysis of many physical and engineering applications. Increases in available computational capabilities have permitted the development and testing of sophisticated models in the numerical simulation of turbulent flows. Direct numerical simulation, where all essential scales of the turbulent flow are resolved by solving the unsteady Navier-Stokes equations, are possible only at low to moderate Reynolds numbers. Turbulent flow analysis for engineering applications, therefore, can only be achieved by utilizing the time-averaged Navier-Stokes equations coupled with some level of modelling.

The complex structure of swirling flowfields requires careful consideration of the turbulence model derivation and development. The analysis of turbulent transport and modeling evolves from the Reynolds-averaged Navier-Stokes equations and auxiliary equations for velocity and length scales for eddy viscosity specifications. Simple eddy viscosity models based on the Boussinesq hypothesis of linear relationship between turbulent shear stress and rate of strain have been quite successful in predicting a wide variety of turbulent flows.

One of the widely used models is the two-equation $k-\epsilon$ model. The model developed originally by Launder and Spalding⁽¹⁾ was successful in providing good predictions for a large range of turbulent flows. The equations can be derived from the full transport equations for the Reynolds stresses assuming fully turbulent flow. Effects such as that of rotation which are included in the Reynolds stress equations are cancelled out and the resulting scalar $k-\epsilon$ equations are invariant to system rotation.

For low-Reynolds number flows close to solid boundaries, adjustments to the model are needed to bridge the viscous dominated sublayer region with the fully turbulent flow region. The success of the wall function method depends on the universality of the turbulent structure near the wall. In many complex flows, however, the flowfield near the wall has to be determined accurately and the traditional wall-function method is not satisfactory. This is because the specification of all turbulence quantities in terms of the friction velocity fail at separation where the flow near the wall is no longer controlled by the wall shear stress. Patel et al.⁽⁶⁾ assessed the relative performance of various models which describe the near-wall flows and found that there are still areas of improvements needed to accurately model flow behavior near the wall.

Jones and Launder⁽⁷⁾ extended the original $k-\epsilon$ model to the low-Reynolds number form which allowed the calculation to be performed all the way to the wall. Numerical difficulties of accurately resolving the large gradients close to the wall necessitates resolving the wall region with very fine grid structure. Chen and Patel⁽²⁾ introduced a method to resolve the near-wall region which combines the standard $k-\epsilon$ model with the one-equation model of Wolfshtein⁽⁸⁾ near the wall. In this "two-layer" model an algebraically prescribed eddy-viscosity for the wall region is coupled to the $k-\epsilon$ model to describe the details of the flow in the vicinity of the wall. Momentum and continuity equations are solved up to the wall and this reduces the physical uncertainties of near-wall turbulence and the numerical difficulties of resolving the vary large gradients of turbulence parameters.

The purpose of this paper is to discuss the application of the $k-\epsilon$ turbulence model with various near-wall treatments in the prediction of confined swirling flows. These models include, the standard wall function approach (WF), Chen and Patel's⁽²⁾ two-layer model (2L), and Lam and Bremhorst⁽³⁾ low-Reynolds number model (LB).

Evaluation of the various turbulence models was performed by comparison with two selected experimental studies. The first is that of Daily and Nece⁽⁴⁾ where rotating disk cavity circulation and secondary flows are induced by a rotating wall. The second is that of Roback and Johnson⁽⁵⁾ for a confined double concentric jets with a sudden expansion. Flow swirl in this case is induced by imposing a tangential velocity component at the outer jet.

Numerical predictions for turbulent flows in two-dimensional axisymmetric geometries were obtained using a finite-volume second order upwind differencing scheme on a non-staggered grid with a pressure correction method based on the SIMPLE algorithm⁽⁹⁾. The development and evaluation of the turbulence models for rotating flows is part of an ongoing program to assess different models for rotating machinery applications. A discussion on the effects of swirl and streamline curvature on the turbulence structure through the gradient Richardson number formulation is given. Key problem areas will be identified and recommendations for the near-wall treatment as they pertain to rotating flows will be proposed.

MODEL AND EQUATION FORMULATION

Consider an incompressible, statistically steady and axisymmetric turbulent flow, the Reynolds averaged momentum and continuity equations can be expressed in a generalized form as;

$$\frac{\partial(\rho\Phi)}{\partial t} + \frac{\partial(\rho u\Phi)}{\partial x} + \frac{1}{r} \frac{\partial(\rho v r\Phi)}{\partial r} = \frac{\partial}{\partial x}(\Gamma\Phi_x \frac{\partial\Phi}{\partial x}) + \frac{1}{r} \frac{\partial}{\partial r}(r\Gamma\Phi_r \frac{\partial\Phi}{\partial r}) + S_\Phi \quad (1)$$

where Φ is the dependent variable

$\Phi = u, v, w$ for the axial, radial, and tangential velocities

ρ, μ , and S_Φ are the fluid density, viscosity and the source terms for the variable Φ

The source terms for the dependent variables are;

Axial direction, $\Phi=u, \Gamma\Phi_x = 2\mu_e, \Gamma\Phi_r = \mu_e$

$$S_u = -\frac{\partial p}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r}(\mu_e r \frac{\partial v}{\partial x}) \quad (2)$$

Radial direction, $\Phi=v, \Gamma\Phi_x = \mu_e, \Gamma\Phi_r = 2\mu_e$

$$S_v = \frac{\partial}{\partial x}(\mu_e \frac{\partial u}{\partial r}) - 2\mu_e \frac{v}{r^2} + \rho \frac{w^2}{r} - \frac{\partial p}{\partial r} \quad (3)$$

Tangential direction, $\Phi=w, \Gamma\Phi_x = \mu_e, \Gamma\Phi_r = \mu_e$

$$S_w = -\frac{\rho v w}{r} - \frac{w}{r^2} \frac{\partial}{\partial r}(r\mu_e) \quad (4)$$

TURBULENCE MODELS In the two-equation k- ϵ model transport equations for the turbulent kinetic energy (k) and energy dissipation (ϵ) can be written in the same general form as equation (1).

Turbulent Kinetic energy equation

$$\Phi = k, \Gamma\Phi_x = \Gamma\Phi_r = \mu + \frac{\mu_t}{\sigma_k}, \text{ and } S_\Phi = G - \rho\epsilon \quad (5)$$

Energy dissipation equation

$$\Phi = \epsilon, \Gamma\Phi_x = \Gamma\Phi_r = \mu + \frac{\mu_t}{\sigma_\epsilon}, \text{ and } S_\Phi = \frac{\epsilon}{k} (C_1 f_1 G - C_2 f_2 \rho\epsilon) \quad (6)$$

σ_k and σ_ϵ are turbulent Schmidt numbers G denotes the rate of production of the turbulent kinetic energy and is express as;

$$G = \mu_e \left\{ 2 \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial r} \right)^2 + \left(\frac{v}{r} \right)^2 \right] + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial w}{\partial r} - \frac{w}{r} \right)^2 \right\} \quad (7)$$

$$\mu_e = \mu + \mu_t, \text{ and the eddy viscosity is obtained from } \mu_t = C_\mu f_\mu \rho \frac{k^2}{\epsilon}$$

$C_\mu, C_1, C_2, \sigma_k$ and σ_ϵ are constants whose values are 0.09, 1.44, 1.92, 1.0, 1.0, respectively.

Near-Wall Treatment A near-wall turbulent flow can be divided into two regions, the inner viscous sublayer where low turbulence Reynolds number effects are important and the velocities decrease rapidly to zero at the wall, and an outer fully turbulent region. The successful use of the k-ε turbulence model for many complex flows depends on how accurately the flowfield near the wall is determined. Different models are used to treat this thin sublayer region, they include:

Wall function methods, the following equations are assumed to hold

$$u^+ = y^+ \quad \text{for } y^+ \leq 11.6 \quad (8)$$

$$u^+ = \frac{1}{k} \ln(Ey^+) \quad \text{for } y^+ > 11.6 \quad (9)$$

where $u^+ = \frac{u}{u_\tau}$, $y^+ = \frac{u_\tau y}{\nu}$ and $u_\tau = (\frac{\tau_w}{\rho})^{1/2}$

τ_w is the wall shear stress which is estimated from

$$\tau_w = \frac{\mu u_p}{\delta} \quad \text{for } y^+ \leq 11.6 \quad (10)$$

$$\tau_w = \frac{\kappa C_\mu^{0.25} \rho u_p k^{0.5}}{\ln(EC_\mu^{0.25} \rho \delta k^{0.5}/\mu)} \quad \text{for } y^+ > 11.6 \quad (11)$$

where u_p denotes the velocity component parallel to the wall, and δ is the normal distance from the wall

In this approach, k and ε equations are solved with $f_\mu = f_1 = f_2 = 1$ only in the fully turbulent region beyond some distance from the wall. Boundary conditions, i.e., velocity components and turbulence parameters at that distance are specified in terms of the friction velocity (u_τ).

In the low-Reynolds number model, the flow is resolved all the way to the wall with a very fine mesh. Many models have been proposed that are based on the k-ε model and differ mainly in the choice of the damping functions f_μ , f_1 and f_2 to bridge the gap between the sublayer and the fully turbulent regions. Lam and Bremhorst's model⁽³⁾ is used in this work, where

$$f_\mu = [1 - \exp(-0.016R_y)]^{1/2} (1 + \frac{20.5}{R_t})$$

$$f_1 = 1 + (\frac{0.06}{f_\mu})^3 \quad \text{and} \quad f_2 = 1 - \exp(-R_t^2)$$

$$R_y = \frac{k^{1/2} y}{\nu} \quad \text{and} \quad R_t = \frac{k^2}{\nu \varepsilon} \quad \text{are turbulent Reynolds numbers}$$

These damping functions tend to unity with increasing distance from the wall. In order to resolve the very large gradients of turbulence parameters a fine mesh is required in the viscous sublayer which increases the computational time and numerical difficulties may be encountered.

In order to alleviate some of the problems encountered in the low-Reynolds number approach and yet accurately resolve the near-wall region, Chen and Patel⁽²⁾ pursued the two-layer concept. In this model a simple algebraically prescribed eddy-viscosity model for the wall region is coupled to the k-ε model for the outer flow to describe the details of the flow. Unlike the low-Reynolds number model that requires the solution of transport equations of both k and ε all the way to the wall, the one-equation model requires the solution of only the turbulent kinetic energy equation in the sublayer region while algebraically specifying the eddy-viscosity and energy dissipation.

$$\nu_t = C_\mu k^{1/2} L_\mu \quad \text{and} \quad \varepsilon = k^{3/2}/L_\varepsilon$$

The length scales L_μ and L_ε contain the necessary damping effects in the near-wall region in terms of the turbulence Reynolds number R_y

$$L_\mu = C_1 y [1 - \exp(-R_y/A_\mu)] \quad (12)$$

$$L_\varepsilon = C_1 y [1 - \exp(-R_y/A_\varepsilon)] \quad (13)$$

The length scales L_μ and L_ε become linear and approach $C_1 y$ with increasing distance from the wall. $C_1 = \kappa C_\mu^{-0.75}$ with $A_\varepsilon =$

$2C_1$. Chen and Patel⁽²⁾ gave values for the constant $A_\mu = 70$. The damping effects decay rapidly with distance from the wall

independent of the magnitude of the wall shear stress. The matching between the one-equation and the standard k- ϵ models is carried out along prescribed grid lines where $Re_y \sim 200$.

STREAMLINE CURVATURE AND SWIRL CORRECTIONS Turbulent flows in many engineering applications such as turbomachinery and combustion devices are frequently subjected to complicating influences such as mean strain and body forces due to rotation. In such complex flows streamline curvature and swirl can exert a large influence on the structure of turbulence. Bradshaw⁽¹⁰⁾ reviewed the effects of streamline curvature and discussed the large effect exerted on shear-flow turbulence by curvature of streamlines in the plane of the main shear. So and Mellor⁽¹¹⁾ suggested that the appropriate parameter governing this effect is $F = \frac{u/R}{\partial u/\partial y}$, where R is the radius of streamline curvature. Militzer et al.⁽¹²⁾ provided a simple generalization of this parameter for a 2-D recirculating flow as

$$F = \frac{(u^2 + v^2)^{1/2}/R}{(\partial u/\partial y + \partial v/\partial x)} \quad (14)$$

They modified the turbulence production term G in the turbulent energy equations to include curvature effects and obtained improved predictions. Launder et al.⁽¹³⁾ proposed a simple modification to the constant C_2 in the ϵ -equation to account for streamline curvature due to swirl in the form

$$C_2 = 1.0 - 0.2 R_{is} \quad (15)$$

where R_{is} is a swirl Richardson number defined by

$$R_{is} = \frac{w/r^2 \partial(rw)/\partial r}{(\partial u/\partial r)^2 + (r \partial(w/r)/\partial r)^2} \quad (16)$$

Another expression of R_{is} can also be derived as

$$R_{is} = \frac{k^2}{\epsilon^2} \frac{w}{r^2} \frac{\partial(rw)}{\partial r} \quad (17)$$

The basis of the above correction is that the effect of swirl on turbulence can be modelled through an increase in the length scale of the energetic turbulence eddies.

Abujela and Lilley⁽¹⁴⁾ used a modified C_2 form (Eq. 15) with both definitions of R_{is} from Equations (16) and (17) as applied to turbulent swirling flows. They concluded that Eq. (16) Richardson number gave better comparisons with experiment as compared to Eq. (17) Richardson's number. They also found the value of C_2 obtained from Eq. (15) had to be limited to $0.1 \leq C_2 \leq 2.4$ with C_μ and other constants assigned their conventional values.

Srinivasan and Mongia⁽¹⁵⁾ further split the Richardson number into two parts - the swirl Richardson number R_{is} and the curvature Richardson number R_{ic} and corrected C_2 in the ϵ -equations as:

$$C_2 = 1.92 \exp(2\alpha_s R_{is} + 2\alpha_c R_{ic}) \quad (18)$$

where R_{is} is given by equations (16) or (17) and

$$R_{ic} = \frac{(u^2 + v^2)^{1/2}/R}{\left(\frac{1}{r} \frac{\partial u}{\partial r} + \frac{\partial v}{\partial x}\right)} \quad (19)$$

where R is the radius of curvature given by $R = \frac{(u^2 + v^2)^{3/2}}{uv \left(\frac{1}{r} \frac{\partial r}{\partial r} - \frac{\partial u}{\partial x}\right)}$, and α_s and α_c are constants with values ranging between 0.1 and 2.4.

Chang et al.⁽¹⁶⁾ investigated the streamline curvature effects in the k- ϵ model. They managed to obtain satisfactory results in their hybrid k- ϵ model where modifications of curvature effects in C_2 is made only in regions where the streamline curvature is large.

In the present study curvature and swirl modifications are made to C_2 similar to Eq. (18) of Srinivasan and Mongia with R_{is} as in Eq. (16) and R_{ic} as in Eq. (19). The exponential form ensures that C_2 will never become negative. Numerical testing with several values of α_s and α_c reveal that C_2 may become very large and therefore, had to be limited to $0.1 \leq C_2 \leq 2.4$.

MODEL EVALUATION

The various near-wall treatment models are analyzed by comparing model predictions with experimental data. Two cases of rotating flow experiments were selected for validation, they include; Daily and Nece⁽⁴⁾ for rotating disk cavity experiment and Roback and Johnson⁽⁵⁾ for swirling flow in a confined double concentric jets with a sudden expansion. The main criterion for selecting these cases is the different mechanisms used to generate swirling flows. In Daily and Nece experiment flow rotation is induced by the rotating wall, while in Roback and Johnson's Experiment, swirl is imparted to the flow by an outer swirling jet into a sudden expansion. Different rotation mechanisms affecting turbulence can highlight the differences between various turbulence models and offer certain corrections that would prove useful in accurately analyzing the effects of swirl.

CASE (1) - DAILY AND NECE⁽⁴⁾ In their experimental and analytic study Daily and Nece⁽⁴⁾ analyzed the steady-state turbulent flow in enclosed rotating disk cavities. They characterized the existence of four flow regimes depending on the rotational Reynolds number and cavity aspect ratio. The two-dimensional axisymmetric flow considered is that of an incompressible flow bounded by a disk (rotor) and a stationary end wall (stator) of a chamber as shown in Figure 1. The ratio of the axial clearance between the rotor and the stator (s) to the radius of the disk (a) is 0.0255. The disk rotates with a rotational Reynolds number $R=4.4 \times 10^6$ defined as $R=\Omega a^2/\nu$, where Ω is the disk rotational speed in rad/sec and ν is the kinematic viscosity.

Numerical computations were performed on a 33×75 grid with different grid clustering near the walls for the different near-wall models. Figure 2, shows the velocity vectors at the top region of the cavity using the WF model. Centrifugal forces move the fluid radially outward on the disk, axially away from the disk on the wall casing, and radially inwards on the stationary end wall. Figure 3, shows the axial variations of the radial velocity component (v) at a radial position $r/a=0.765$. The agreement is fair with some discrepancy for all near-wall models close to the rotating disk. Figure 4, shows the axial variation of the tangential velocity (w) component at the radial position. At the rotating disk ($x=0$), the tangential velocity component approach the value ($a\Omega$). The 2L near-wall model seem to offer closer agreement with the data than the other two models.

The presence of corner regions presents a difficulty in defining the normal distances used in the definition of turbulent Reynolds number (R_y). In the present analysis, values of the normal distance from a wall were based on the normal distance to the nearest solid boundary. Streamline curvature and swirl corrections have not been used in this case.

CASE (2) - ROBACK AND JOHNSON⁽⁵⁾ Predictions of the experiments of Roback and Johnson⁽⁵⁾ have been presented by several workers, e.g. Sloan et al.⁽¹⁷⁾ and Durst and Wennergren⁽¹⁸⁾. Unfortunately, inlet profiles were not provided in their experiment. Therefore, calculations were started at the expansion plane using the measured velocity profiles at 5mm downstream of the expansion after some adjustments near the edges of the coaxial jets. Measurements of all three main turbulent intensities were used to calculate inlet values of the turbulent kinetic energy. Energy dissipation rate was estimated from

$$\epsilon = \frac{C_\mu k^{3/2}}{L} \quad (20)$$

where L is a length scale of turbulence at the inlet of the order of $L=10^{-4}$ m.

Figure 5, shows an illustration of the test chamber geometry. The confluence plane of the primary (inner) and secondary (outer) jet streams coincides with the chamber expansion plane. The chamber diameter is about twice the secondary tube diameter. The exit from the 8-bladed, 30° , free vortex swirl generator is located approximately 0.05 m upstream from the confluence plane.

A prevalent phenomenon in axisymmetric swirling flows in such geometries is the "bubble" or vortex breakdown which has been studied extensively^(19,20,21,22). The near axisymmetric breakdown can be partially understood from a simplified analysis of the role of pressure and centrifugal forces. It is identified by a slowly varying vortex core which undergoes an abrupt and rapid deceleration, forming a free stagnation point, followed by a region of flow reversal. It is known that the structure of vortex breakdown is unstable and asymmetric in the azimuthal direction, and displays unsteadiness in the axial direction^(23,24). However, no periodic or nonaxisymmetric behavior attributable to the vortex breakdown was observed in Roback and Johnson's experiment.

In the numerical simulation of the experiment, a 150×100 grid nodes was used with different clustering on the walls for the different near-wall models used. Figure 6, shows the velocity vectors indicating the presence of a closed recirculation zone at the center with additional zones at the corner downstream and between the inner jet and the outward diverted secondary jet. The figure also shows flow diversion outwards with high gradients characterized by large turbulent shear and fluctuation levels.

Comparisons were made of the radial variations of flow variables at two axial locations, $x=0.025$ m upstream of the vortex bubble and $x=0.102$ m located inside the vortex bubble. Figure 7a, shows the radial variation of the axial velocity

profile at $x=0.025$ m using the WF method, 2L model and LB model. Fair agreement is predicted by the different models. They also seem to predict small negative velocities at a radial position $r \sim 0.0153$ m (the interface between the inner and outer jets), slightly underpredicting its strength and width. Figure 7b, shows the radial variation of the axial velocity profiles at $x=0.102$ m. The 2L model shows a better agreement with the experimental data. These velocities are slightly underpredicted above the outer jet diameter.

Radial variations of the tangential velocities are shown in Figure 8a and 8b at $x=0.025$ m and $x=0.102$ m respectively. Figure 8a shows that the 2L model offers a better agreement with the experiment as compared with the WF and LB models. At $x=0.102$ m, Figure 8b shows that the swirl velocity is underpredicted. That is because the radial transfer of circumferential velocity is highly dependent on the turbulent diffusion mechanisms which are not accurately modelled in the isotropic eddy-viscosity $k-\epsilon$ model used here.

The turbulent intensity predictions for the $k-\epsilon$ model using the different near-wall treatments seem to follow similar trends as shown in Figures 9(a,b), 10(a,b), and, 11(a,b). In general within the approximations of the isotropic $k-\epsilon$ model, the 2L model offer a marginal improvements over the WF and LB near wall models. The peaks in the axial, radial, and tangential turbulence intensities occur around the edges of the inner and outer jets. Figure 13a, shows the axial-azimuthal Reynolds stress profile at $x=0.025$ m. Figure 12b and 12c, show the axial-radial and radial-azimuthal Reynolds stress profiles at $x=0.102$ m.

The analysis of the main turbulent intensities and of the Reynolds stress components using the isotropic eddy-viscosity $k-\epsilon$ turbulent model do not reveal exclusively the advantage of one near-wall model over the other. Moreover, Reynolds shear stress profiles are sensitive to the upstream inlet conditions and the developing mean flowfield. Although the mean flow quantities show a general trend of improved predictions using the 2L near-wall model, the main effects of turbulence are due to anisotropy of Reynolds stresses especially around the highly sheared region of the outward diverted outer jet and the vortex bubble.

Streamline curvature and swirl corrections have been attempted in the present analysis with little success. Corrections of C_2 using equation (18) with equations (16) and (19) for the swirl and curvature Richardson numbers. Figure 13(a,b) show the radial distribution of the radial velocity at $x=0.025$ m, and the axial velocity at $x=0.102$ m. Small improvement is detected with these corrections. The constants α_s and α_c used are those recommended by Srinivasan and Mongia⁽¹⁵⁾ in their calculations ($\alpha_s=-0.75$ and $\alpha_c=-2.0$). These constants were not optimized in the present calculations.

CONCLUSIONS

Flow predictions were performed for the standard $k-\epsilon$ turbulence model with different near-wall treatments to assess their performance when applied to rotating flows. Comparisons of predictions with the experimental data of Daily & Nece, and Roback & Johnson show reasonable agreement for all near-wall models and in general, the two-layer model seem to offer better comparisons compared to the wall function and Lam & Bremhorst low-Reynolds number models. From a computational perspective, the two-layer model require less computer time and relatively few grid points in the wall region than the low-Reynolds number model and is less sensitive to the location of the interface between the sublayer and the fully turbulent regions. Streamline curvature and swirl corrections show small improvements. However, further study is needed to optimize their constants.

REFERENCES

- (1) Launder, B. E. and Spalding, D. B. "The Numerical Computation of Turbulent Flows", Computer Methods in Applied Mechanics and Engineering, Vol. 3, 1974, pp. 269-289
- (2) Chen, H. C. and Patel, V. C. "Near-Wall Turbulence Models for Complex Flows Including Separation", AIAA Journal, vol. 26, No. 6, 1988, pp. 641-648
- (3) Lam, C. and Bremhorst, K. "A Modified Form of the $k-\epsilon$ Model for Predicting Wall Turbulence", Trans. ASME J. Fluids Eng., Vol. 103, 1981, pp. 456-460
- (4) Daily, J. W. and Nece, R. E. "Chamber Dimension Effects on Induced Flow and Frictional Resistance of Enclosed Rotating Disks", Trans. ASME J. Basic Eng., 1960, pp. 217-232
- (5) Roback R., and Johnson, B. "Mass and Momentum Turbulent Transport Experiment With Confined Swirling Co-Axial Jets", NASA CR-168252, 1983
- (6) Patel, V. C., Rodi, W., and Scheuerer, G. "Turbulence Models for Near-Wall and Low Reynolds Number Flows: A Review", AIAA Journal, Vol. 23, No. 9, 1985, pp. 1308-1319
- (7) Jones, W. P., and Launder, B. E. "The Calculation of Low-Reynolds-Number Phenomena With a Two-Equation Model of Turbulence", Int. J. Heat & Mass Transfer, vol.16, 1973, pp. 1119-1130
- (8) Wolfshtein, M. "The Velocity and Temperature Distribution in One-Dimensional Flow With Turbulence Augmentation and Pressure Gradients" Int. J. Heat & Mass Transfer, vol. 12, 1969, pp. 301-318

- (9) Patankar, S. V., "Numerical Heat Transfer and Fluid Flow", McGraw-Hill, New York, 1980
- (10) Bradshaw, p. "Effects of Streamline Curvature on Turbulent Flow" AGARDograph No. 169, 1973
- (11) So, R. and Mellor, G. "Experiments on Convex Curvature Effects in Turbulent Boundary Layers", J. Fluid Mech., Vol. 60, 1973, pp. 43-62
- (12) Militzer, J., Nicoll, W., and Alpay, S. "Some Observations on the Numerical Calculations of the Recirculation Region of Twin Parallel Symmetric Jet Flow", Symp. on Turbulent Shear Flows, University Park, Pennsylvania, April 18-20, 1977
- (13) Launder, B., Priddin, C., and Sharma, B. "The Calculation of Turbulent Boundary Layer on Spinning and Curved Surfaces", ASME J. Fluids Eng. 1977, pp. 231-239
- (14) Abujela, M. T., and Lilley, D. G. "Limitations and Empirical Extensions of the k- ϵ Model as Applied to Turbulent Confined Swirling Flows", Paper AIAA-84-0441, Reno, Nevada, Jan. 9-12, 1984
- (15) Srinivasan, R., and Mongia, H. "Numerical Computation of Swirling Recirculating Flows: Final Report" NASA CR-165196, 1980
- (16) Chang, K. C., Chen, C. S., and Uang, C. I. "A Hybrid k- ϵ Turbulence Model of Recirculating Flow" Int. J. for Numerical Methods in Fluids, Vol. 12, 1991, pp. 369-382
- (17) Sloan, D. G., Smith, P. J., and Smoot, L. D. "Modelling of Swirl in Turbulent Flow Systems" Prog. Energy Combust. Sci., Vol. 12, 1986, pp. 163-250
- (18) Durst, F., and Wennerberg, D. "Numerical Aspects of Calculation of Confined Swirling Flows With Internal Recirculation" Int. J. for Numerical Methods in Fluids, Vol. 12, 1991, pp. 203-224
- (19) So, K. L. "Vortex Phenomena in a Conical Diffuser" AIAA J., Vol. 5, 1967, pp. 1072-1078
- (20) Garg, A. K., and Leibovich, S. "Spectral Characteristics of Vortex Breakdown Flowfields", Physics Fluids, Vol. 22, 1979, pp. 2053-2064
- (21) Faler, J. H., and Leibovich, S. "Disrupted States of Vortex Flow and Vortex Breakdown", Physics Fluids, Vol. 20, 1977, pp. 1385-1400
- (22) Escudier, M. P. and Zehnder, N. "Vortex Flow Regimes" J. Fluid Mech., Vol. 115, 1982, pp. 105-121
- (23) Sarpkaya, T. "On Stationary and Travelling Vortex Breakdowns" J. Fluid Mech., Vol. 45, 1971, pp. 545-559
- (24) Faler, J. H., and Leibovich, S. "An Experimental Map of the Internal Structure of a Vortex Breakdown" J. Fluid Mech., Vol. 86, 1978, pp. 313-335

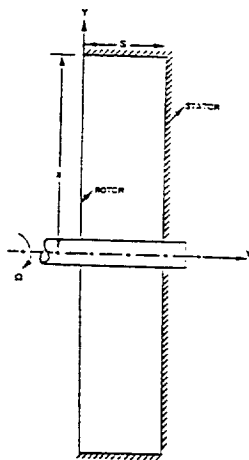


Fig. 1 Rotating Closed Cavity



Fig. 2 Velocity Vectors

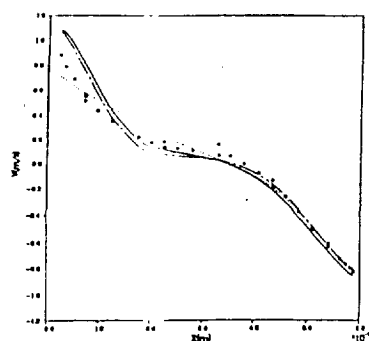


Fig. 3 Axial Distribution of Radial Velocity (m/s), at $r/a=0.765$

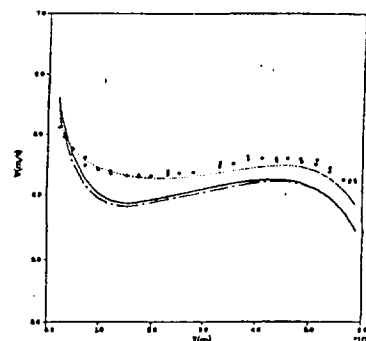
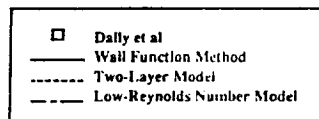


Fig. 4 Axial Distribution of Tangential Velocity (m/s), at $r/a=0.765$

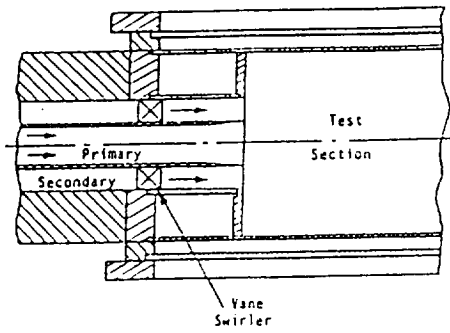


Fig. 5 Swirling Coaxial Jets Discharging into an Expanded Duct

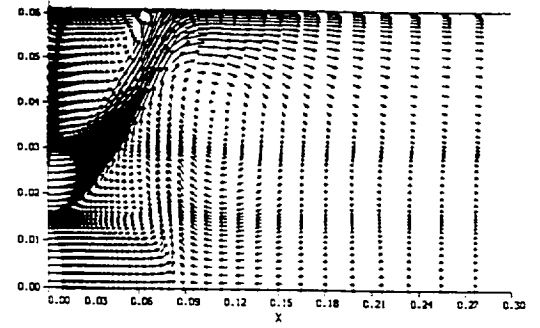


Fig. 6 Velocity Vectors

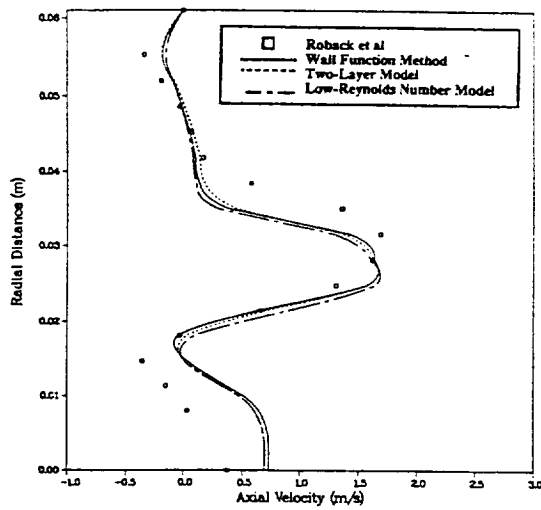


Fig. 7a Radial Distribution of Axial Velocity (m/s) at $x=0.025m$

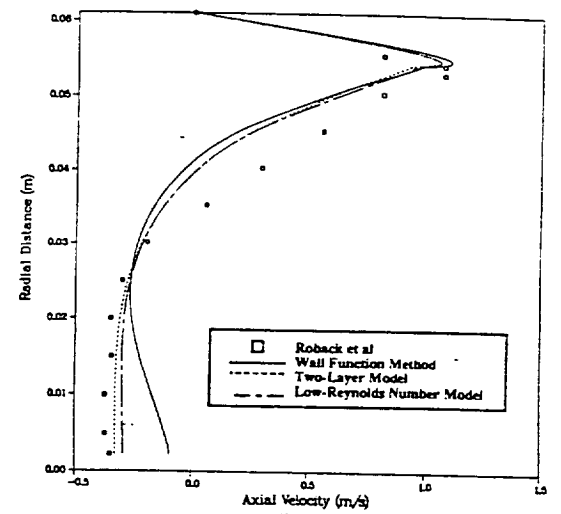


Fig. 7b Radial Distribution of Axial Velocity (m/s) at $x=0.102m$

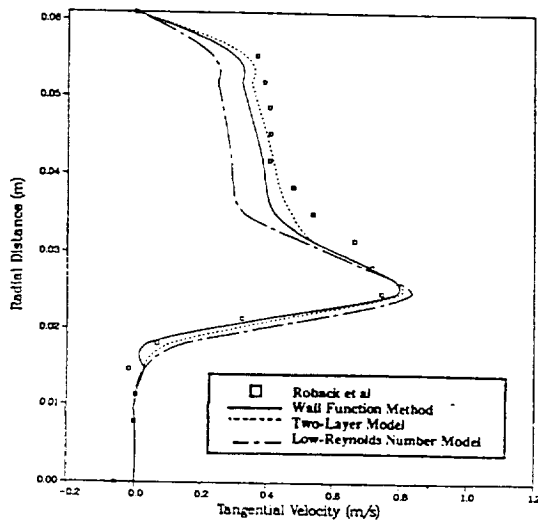


Fig. 8a Radial Distribution of Tangential Velocity (m/s) at $x=0.025m$

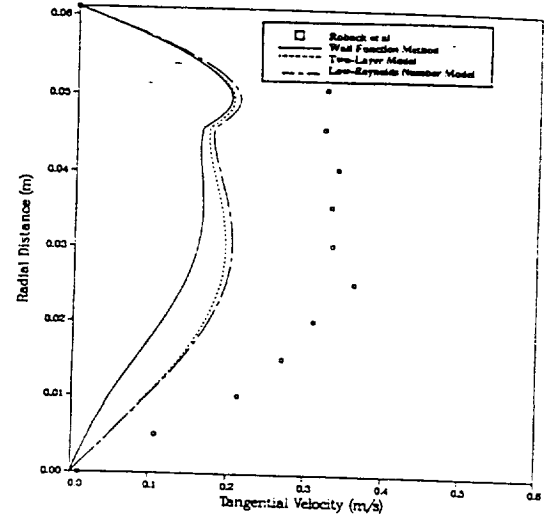


Fig. 8b Radial Distribution of Tangential Velocity (m/s) at $x=0.102m$

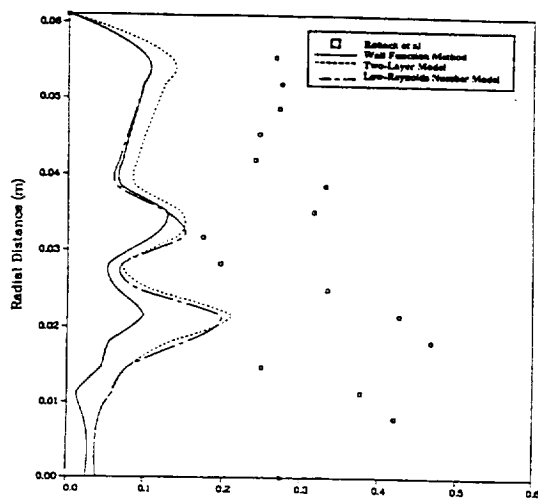


Fig. 9a Radial Distribution of Axial Turbulence Intensity at $x=0.025m$

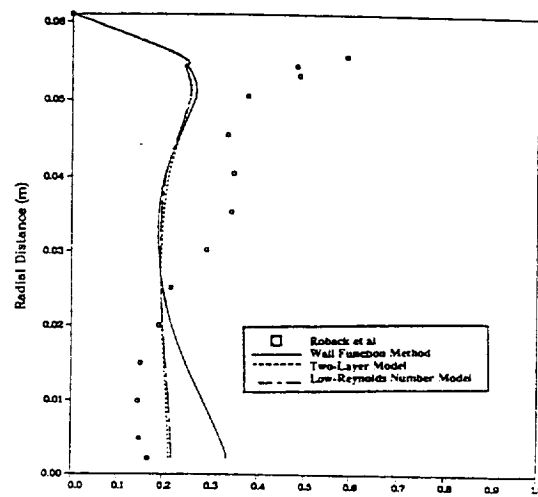


Fig. 9b Radial Distribution of Axial Turbulence Intensity at $x=0.102m$

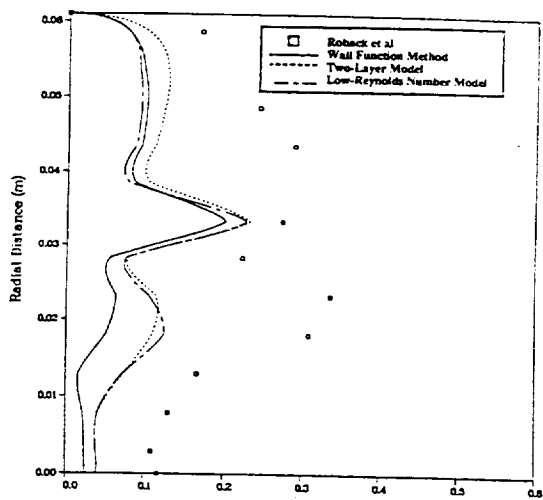


Fig. 10a Radial Distribution of Radial Turbulence Intensity at $x=0.025m$

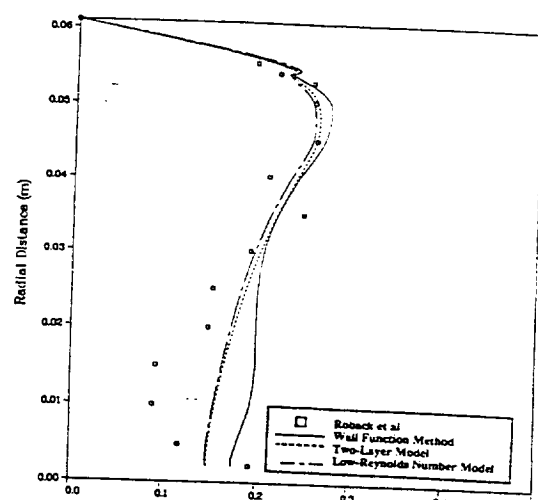


Fig. 10b Radial Distribution of Radial Turbulence Intensity at $x=0.102m$

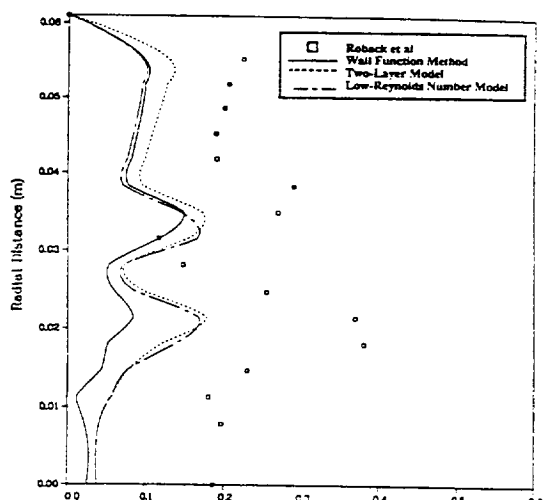


Fig. 11a Radial Distribution of Tangential Turbulence Intensity at $x=0.025m$

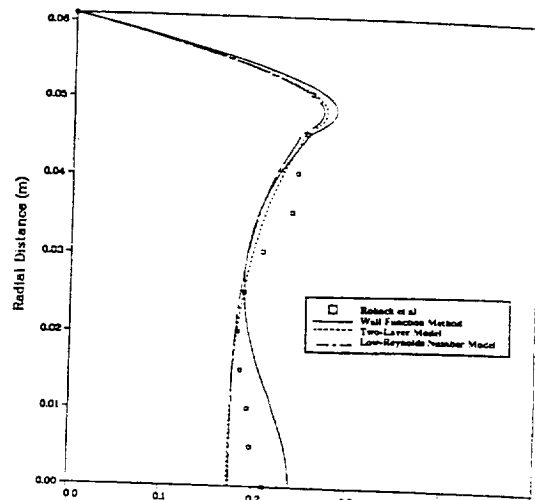


Fig. 11b Radial Distribution of Tangential Turbulence Intensity at $x=0.102m$

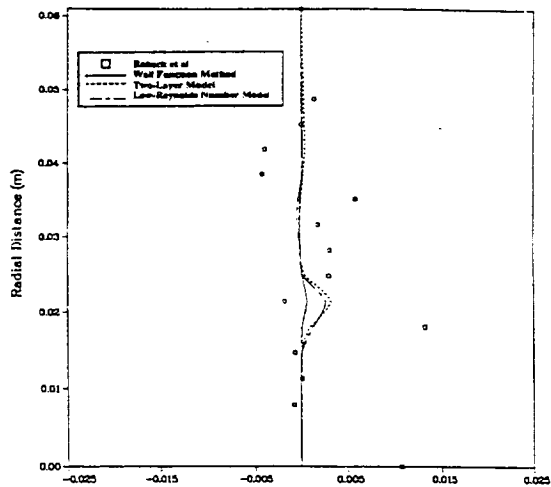


Fig. 12a Radial Distribution of Axial-Azimuthal Reynolds Stresses at $x=0.025\text{m}$

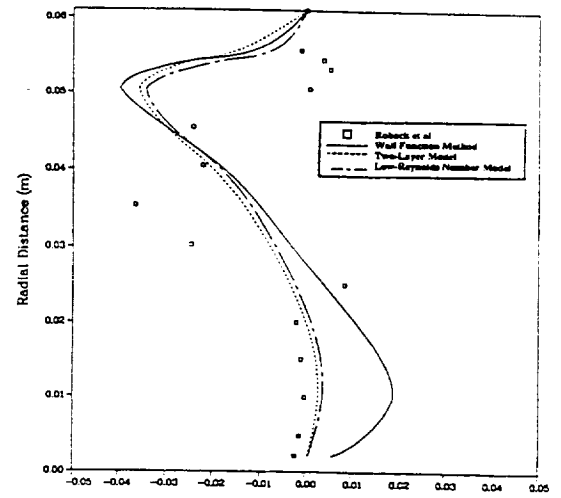


Fig. 12b Radial Distribution of Axial-Radial Reynolds Stresses at $x=0.102\text{m}$

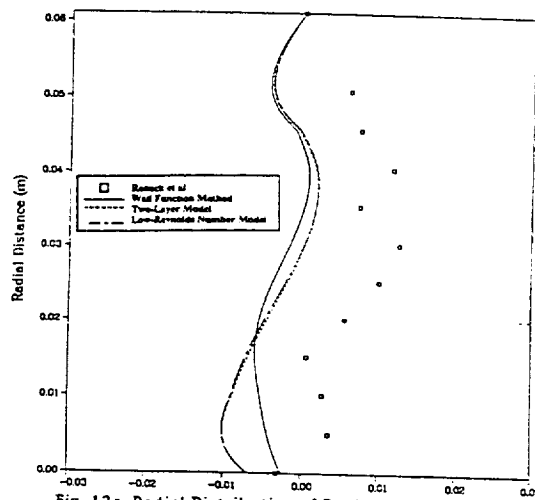


Fig. 12c Radial Distribution of Radial-Azimuthal Reynolds Stress at $x=0.102\text{m}$

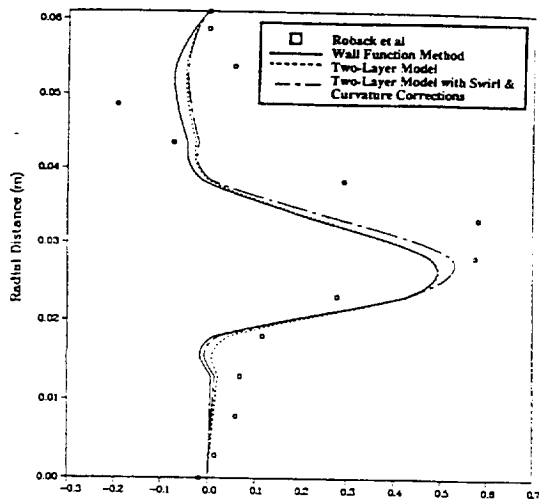


Fig. 13a Radial Distribution of Radial Velocity at $x=0.025\text{m}$

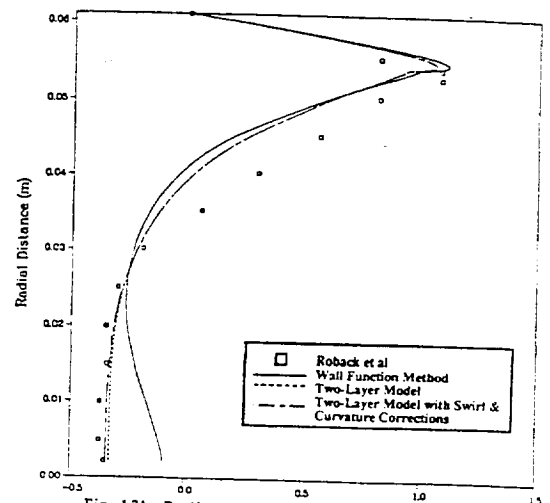


Fig. 13b Radial Distribution of Axial Velocity at $x=0.102\text{m}$

ADVANCED TURBULENCE MODELS FOR TURBOMACHINERY

Ali H. Hadid, Michele E. DeCroix, and Munir M. Sindir
Rocketdyne Division, Rockwell International

ABSTRACT

Development and assessment of the single-time-scale $k-\epsilon$ turbulence model with different near-wall treatments and the multi-scale $k-\epsilon$ turbulence model for rotating flows are presented. These turbulence models are coded as self-contained module decks that can be interfaced with a number of CFD main flow solvers. For each model, a stand-alone module deck with its own formulation, discretization scheme, solver and boundary condition implementations is presented. These satellite decks will take as input (from a main flow solver) the velocity field, grid, boundary condition specifications and will deliver turbulent quantities as output. These modules were tested as a separate entities and although many logical and programming problems were overcome only wider use and further testing can render the modules sufficiently "fool proof".

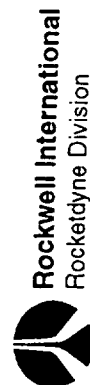
DEVELOPMENT OF A MODULAR FORMAT FOR GENERAL USE TURBULENCE MODELS

**Ali H. Hadid
Michele E. DeCroix**

Rockwell International, Rocketdyne Division

**Workshop for Computational Fluid Dynamic
Applications in Rocket Propulsion**

**April 20-22, 1993
NASA Marshall Space Flight Center**

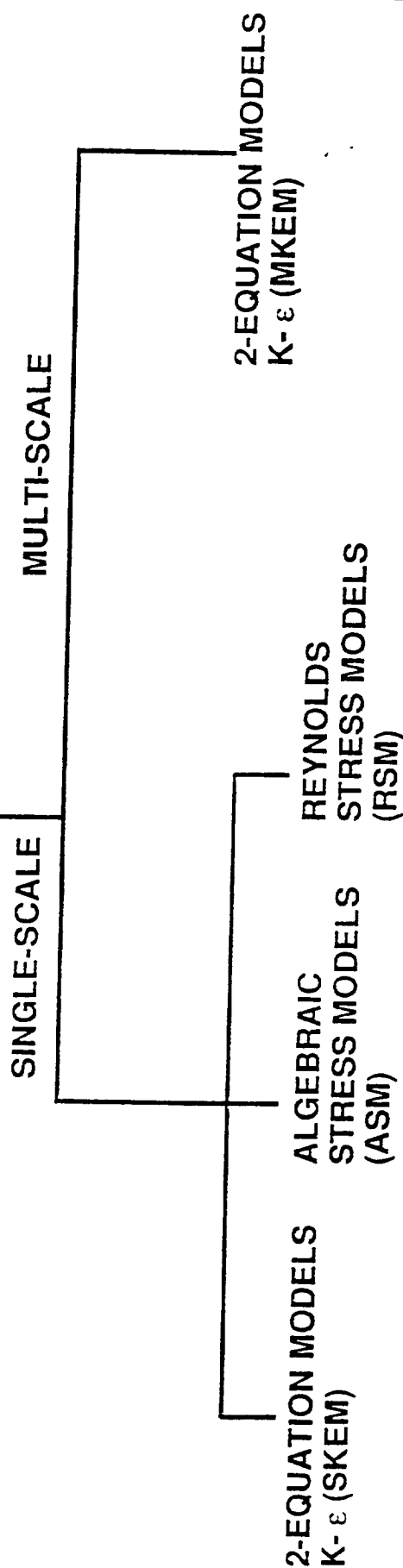


STRUCTURE OF FUTURE CODES LEADS TO DEVELOPMENT OF MODULES

- **FUTURE CODES COMPOSED AT TIME OF EXECUTION**
 - INTELLIGENT DRIVERS
 - MODULES FOR PHYSICAL MODELS
- **INTEGRAL PRE- AND POSTPROCESSING TOOLS**
- **COMMON DATA BASE**

TURBULENCE MODELS TO BE ASSESSED

PHENOMENOLOGICAL SINGLE POINT CLOSURE MODELS



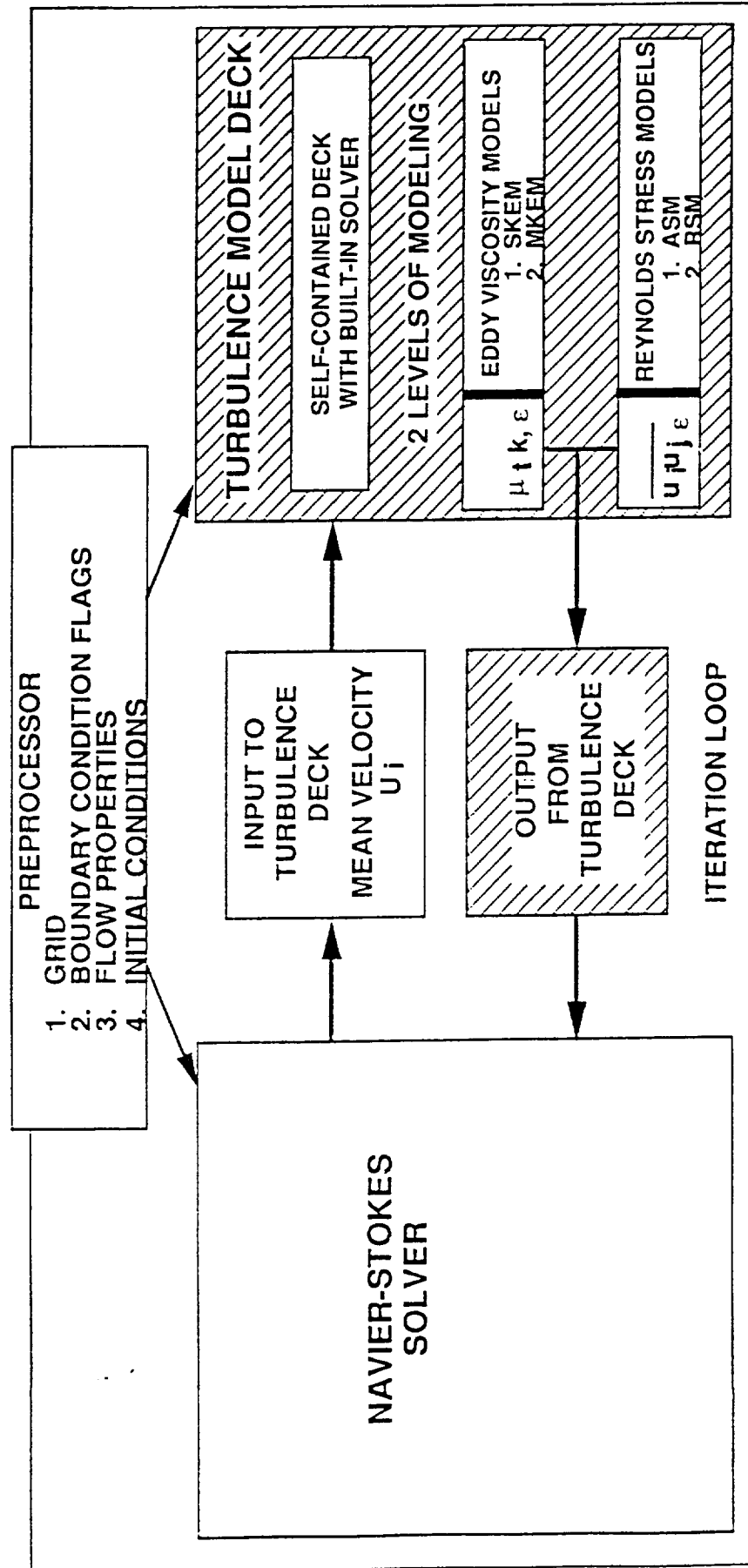
- THE 2-D/AXISYMMETRIC SINGLE-SCALE $K-\epsilon$ MODULE DECK (KEMOD-1) AND THE 2-D/AXISYMMETRIC MULTI-SCALE $K-\epsilon$ MODULE DECK (KEMOD-2) ARE COMPLETE
- NEAR-WALL TREATMENTS WILL INCLUDE (WHERE APPROPRIATE) WALL FUNCTIONS, MULTILAYER MODELS, AND LOW-REYNOLDS NUMBER APPROXIMATIONS



Rockwell International
Rocketdyne Division

TURBULENCE MODEL DECK STRUCTURE AND INTEGRATION WITH NAVIER-STOKES SOLVER

- MODULES ARE BASED ON THE PHENOMENOLOGICAL SINGLE POINT TURBULENCE CLOSURE MODELS
- THEY ARE STRUCTURED BASICALLY TO ACCEPT AS INPUT THE MEAN FLOW VELOCITIES FROM A N-S SOLVER AND TO RETURN TURBULENCE QUANTITIES TO THE SOLVER



USER PROVIDED

ROCKETDYNE PROVIDED

SINGLE-SCALE k-ε MODEL

- GENERALIZED TRANSPORT EQUATION IN 2-D/AXISYMMETRIC GEOMETRY

$$\frac{\partial(\rho u \phi)}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r} (r \rho v \phi) = \frac{\partial}{\partial x} \left(\Gamma \phi_x \frac{\partial \phi}{\partial x} \right) + \frac{1}{r} \frac{\partial}{\partial r} \left(r \Gamma \phi_r \frac{\partial \phi}{\partial r} \right) + S \phi$$

- U-MOMENTUM $\phi = u, \Gamma \phi_x = 2\mu_e, \Gamma \phi_r = \mu_e$

$$S_u = -\frac{\partial P}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r} \left(\mu_e r \frac{\partial v}{\partial x} \right)$$

- V-MOMENTUM $\phi = v, \Gamma \phi_x = \mu_e, \Gamma \phi_r = 2\mu_e$

$$S_v = -\frac{\partial P}{\partial x} - \frac{\partial}{\partial r} \left(\mu_e r \frac{\partial u}{\partial x} \right) - 2\mu_e \frac{v}{r^2} + \frac{\rho w^2}{r}$$

SINGLE-SCALE k-ε MODEL (CONT'D)

• W-MOMENTUM

$$\phi = w, \Gamma\phi_x = \mu_e, \Gamma\phi_r = \mu_e$$

$$S_w = -\frac{\rho v w}{r} + \frac{w}{r^2} \frac{\partial}{\partial r} (r \mu_e)$$

• TURB. KINETIC ENERGY

$$\phi = K, \Gamma\phi_x = \mu + \frac{\mu_t}{\sigma_K} = \Gamma\phi_r$$

$$S_\phi = G - \rho \varepsilon$$

• TURB. ENERGY DISSIPATION

$$\phi = \varepsilon, \Gamma\phi_x = \mu + \frac{\mu_t}{\sigma_K} = \Gamma\phi_r$$

$$S_\phi = \frac{\varepsilon}{K} (c_1 f_1 G - c_2 f_2 \rho \varepsilon)$$



Rockwell International
Rockaldyne Division

CFD 93 013-008/D1/A/H

SINGLE-SCALE k - ϵ MODEL (CONT'D)

$$\bullet \quad G = \mu_e \left\{ 2 \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial r} \right)^2 + \left(\frac{v}{r} \right)^2 \right] + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial w}{\partial x} \right)^2 + \left(\frac{\partial w}{\partial r} - \frac{w}{r} \right)^2 \right\}$$

$$\bullet \quad \mu_t = C_\mu f_\mu \rho \frac{k^2}{\epsilon}$$

$$\mu_e = \mu + \mu_t$$

$$C_\mu = 0.09, \quad c_1 = 1.44, \quad c_2 = 1.92, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.0$$

KEMOD-1 MODULE DECK

- KEMOD-1 IS THE SINGLE-SCALE K- ϵ TURBULENCE MODULE DECK. IT CONSISTS OF TWO SEPARATE ROUTINES KEMOD AND MODIFY WHICH HAVE TO BE LINKED TO THE MAIN FLOW SOLVER
- KEMOD IS CALLED WITHIN THE ITERATION LOOP OF THE MAIN FLOW SOLVER
- THE MEAN VELOCITIES AND OTHER VARIABLES ARE PASSED TO THE MODULE THROUGH ITS ARGUMENT LIST (EXPLAINED IN THE USER'S MANUAL)
- A NONSTAGGERED BODY FITTED GRID ARRANGEMENT IS USED BY THE MODULE. IT USES THE MEAN FLOW VARIABLES (VELOCITIES AND MASS FLUXES) TO CONSTRUCT THE DISCRETIZED ALGEBRAIC EQUATION
- DISCRETIZED ALGEBRAIC EQUATIONS ARE SOLVED BY STONE'S STRONGLY IMPLICIT SOLVER



Rockwell International
Rocketdyne Division

KEMOD-1 MODULE DECK (CONT'D)

- **SUBROUTINE GRIDG**

READS GRID NODE LOCATIONS PASSED FROM MAIN SOLVER AND FOR THE FIRST ITERATION CALCULATES GRID RELATED QUANTITIES (CELL AREAS AND VOLUME, NORMAL DISTANCES FROM SOLID BOUNDARIES AND INTERPOLATION FACTORS)

- **SUBROUTINE CALCE**

ASSEMBLES THE COEFFICIENTS AND SOURCE TERMS FOR THE DISCRETIZED K AND ϵ TRANSPORT EQUATIONS IN THE FORM

$$A_p \phi_p = \sum_{i=E,W,N,S} A_i \phi_i + S \phi$$

THE SUBROUTINE SOLVES THE DISCRETIZED EQUATIONS AFTER MODIFYING THE SOURCES AND BOUNDARY CONDITIONS FOR THE PARTICULAR PROBLEM



Rockwell International
Rocketdyne Division

CFD 93 013 01001/1/11

KEMOD-1 MODULE DECK (CONT'D)

- SUBROUTINE TWOLAY

CALLED IF THE TWO-LAYER OR THE LOW-REYNOLDS NUMBER MODELS ARE USED. IT CALCULATES THE COEFFICIENTS NEEDED TO DESCRIBE THE ENERGY DISSIPATION AND EDDY VISCOSITIES CLOSE TO A WALL

- SUBROUTINE SOLSIP

SOLVES THE SYSTEM OF ALGEBRAIC K AND ϵ EQUATIONS USING STONE'S STRONGLY IMPLICIT METHODS

- SUBROUTINE USERM

THIS SUBROUTINE HAS DIFFERENT ENTRY SECTIONS WHERE VARIABLES ARE UPDATED AND BOUNDARY CONDITIONS ARE SET

- SUBROUTINE MODIFY

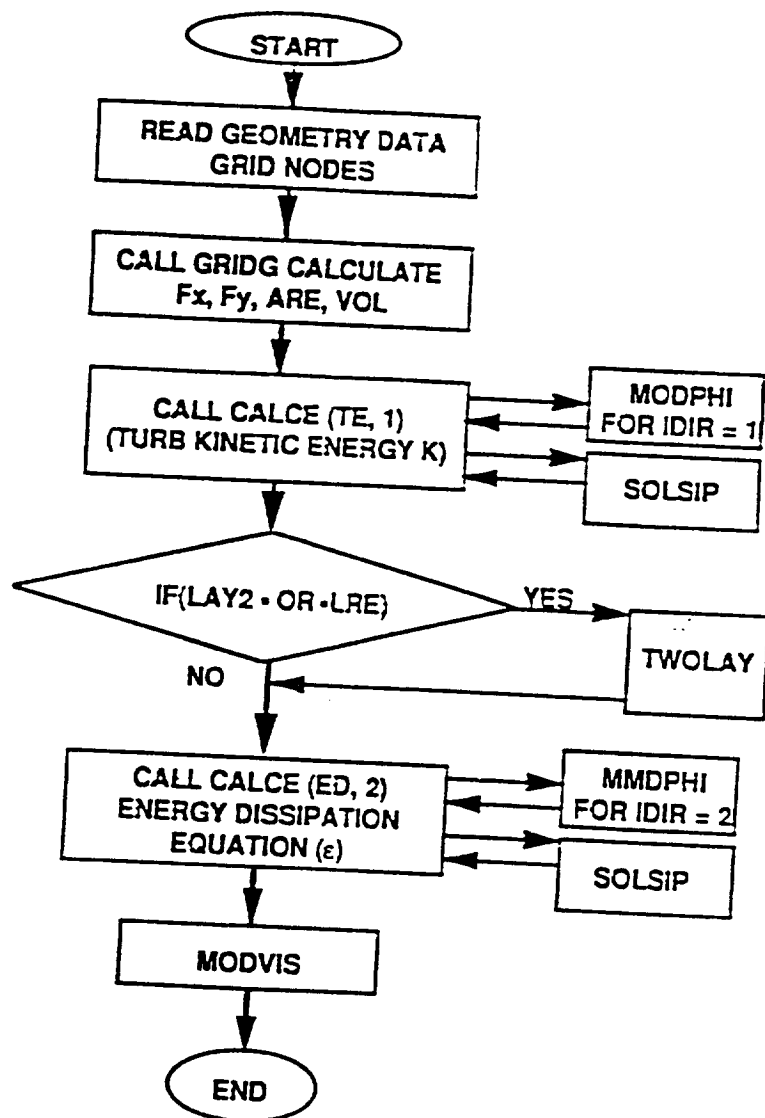
THIS IS THE ONLY SUBROUTINE THAT HAS TO BE CALLED FROM THE MOMENTUM EQUATION SOLVER OF THE MAIN ROUTINE. IT UPDATES THE FLUX SOURCE TERM OF THE DISCRETIZED MOMENTUM EQUATION DUE TO WALL SHEAR STRESSES



Rockwell International
Rocketdyne Division

CFD 93 013 011/D1/W/H

KEMOD-1 MODULE DECK (CONT'D)



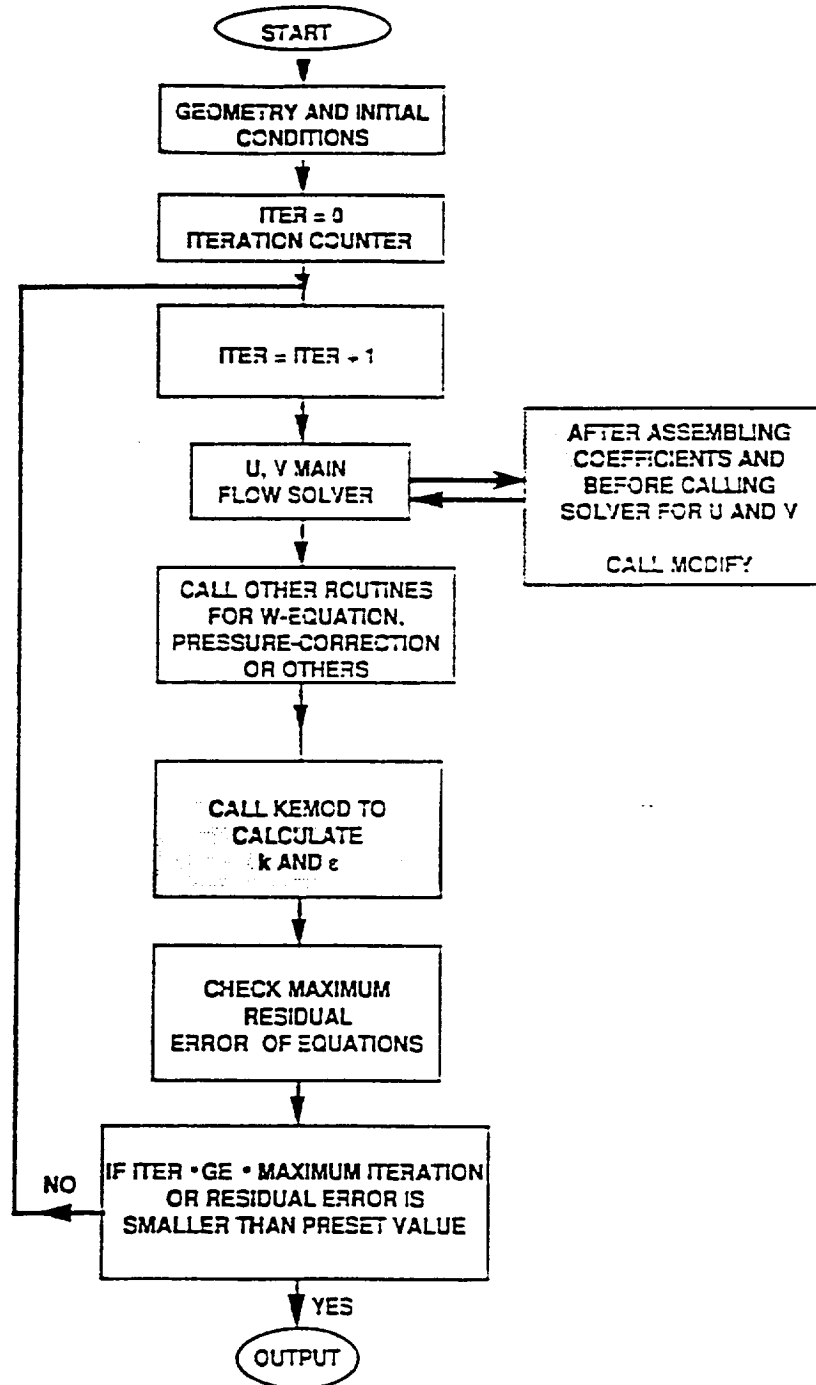
KEMOD FLOW CHART



Rockwell International
Rocketdyne Division

CFD 93 013-012/D1/A4H

KEMOD-1 MODULE DECK (CONT'D)



KEMOD-1 INTERFACE WITH A MAIN SOLVER



Rockwell International
Rocketdyne Division

CFD 93 013-013/D1/AHH

MULTI-SCALE k-ε MODEL

- DERIVED BY PARTITIONING THE TURBULENT ENERGY SPECTRUM INTO TWO SETS OF WAVE NUMBER REGIONS (PRODUCTION AND DISSIPATION RANGES) GIVING TWO EVOLUTION EQUATIONS FOR EACH REGION
- PARTITION LOCATION IS DETERMINED AS PART OF THE SOLUTION
- TURBULENT KINETIC ENERGY IN THE PRODUCTION RANGE OF THE SPECTRUM

$$\phi = k_p, \Gamma\phi_x = \Gamma\phi_r = \mu + \frac{\mu_t}{\sigma k_p}$$

$$S_{kp} = G - \rho\epsilon_p$$



Rockwell International
Rockel-dyne Division

MULTI-SCALE k-ε MODEL (CONT'D)

- ENERGY TRANSFER RATE IN THE PRODUCTION RANGE OF THE SPECTRUM

$$\phi = \varepsilon_p, \quad \Gamma\phi_x = \Gamma\phi_r = \frac{\mu_t}{\sigma k_p}$$

$$S_{\varepsilon_p} = \frac{1}{\rho} C_{p1} \frac{G^2}{K_p} + C_{p2} \frac{G\varepsilon_p}{k_p} - \rho C_{p3} \frac{\varepsilon_p^2}{k_p}$$

- TURBULENT KINETIC ENERGY IN THE DISSIPATION RANGE OF THE SPECTRUM

$$\phi = k_t, \quad \Gamma\phi_x = \Gamma\phi_r = \frac{\mu_t}{\sigma k_t}$$

$$S_{k_t} = \rho\varepsilon_p - \rho\varepsilon_t$$

MULTI-SCALE k-ε MODEL (CONT'D)

- ENERGY DISSIPATION RATE IN THE DISSIPATION RANGE

$$\phi = \varepsilon_t, \Gamma \phi_x = \Gamma \phi_r = \mu + \frac{\mu_t}{\sigma \varepsilon_t}$$

and

$$S_{\varepsilon_t} = \rho C_{t1} \frac{\varepsilon_p^2}{k_t} + \rho C_{t2} \frac{\varepsilon_t \varepsilon_p}{k_t} - \rho C_{t3} \frac{\varepsilon_t^2}{k_t}$$

- MODEL IS SIMILAR TO THAT USED BY KIM AND CHEN WITH CONSTANTS

$$\sigma_{kp} = 0.75, \sigma_{\varepsilon p} = 1.15, \sigma_{kt} = 0.75, \sigma_{\varepsilon t} = 1.15$$

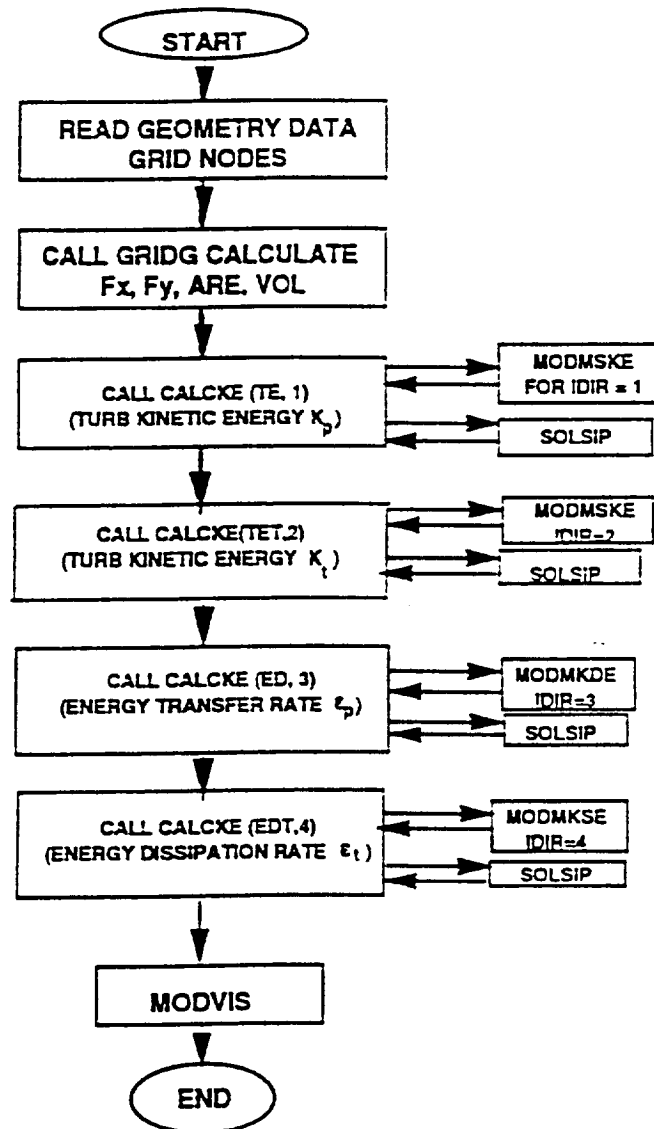
$$C_{p1} = 0.21, C_{p2} = 1.24, C_{p3} = 1.84, C_{t1} = 0.29$$

$$C_{t2} = 1.28, C_{t3} = 1.66 \text{ and } C_{\mu} = 0.09$$

KEMOD-2 MODULE DECK

- KEMOD-2 IS A MULTI-TIME SCALE K- ϵ TURBULENCE MODULE DECK. IT CONSISTS OF TWO MAIN ROUTINES KEMOD AND MODIFY
- KEMOD IS CALLED WITHIN THE ITERATION LOOP OF THE MAIN FLOW SOLVER
- MEAN VELOCITIES AND OTHER VARIABLES ARE PASSED TO THE MODULE THROUGH ITS ARGUMENT LIST (EXPLAINED IN THE USER'S MANUAL
- THE MODULE IS STRUCTURED IN A SIMILAR WAY TO KEMOD-1 AND SUBROUTINE CALCE ASSEMBLES THE COEFFICIENTS AND SOURCE TERMS FOR THE DISCRETIZED K_p , ϵ_p , K_t , ϵ_t TRANSPORT EQUATIONS

KEMOD-2 MODULE DECK (CONT'D)



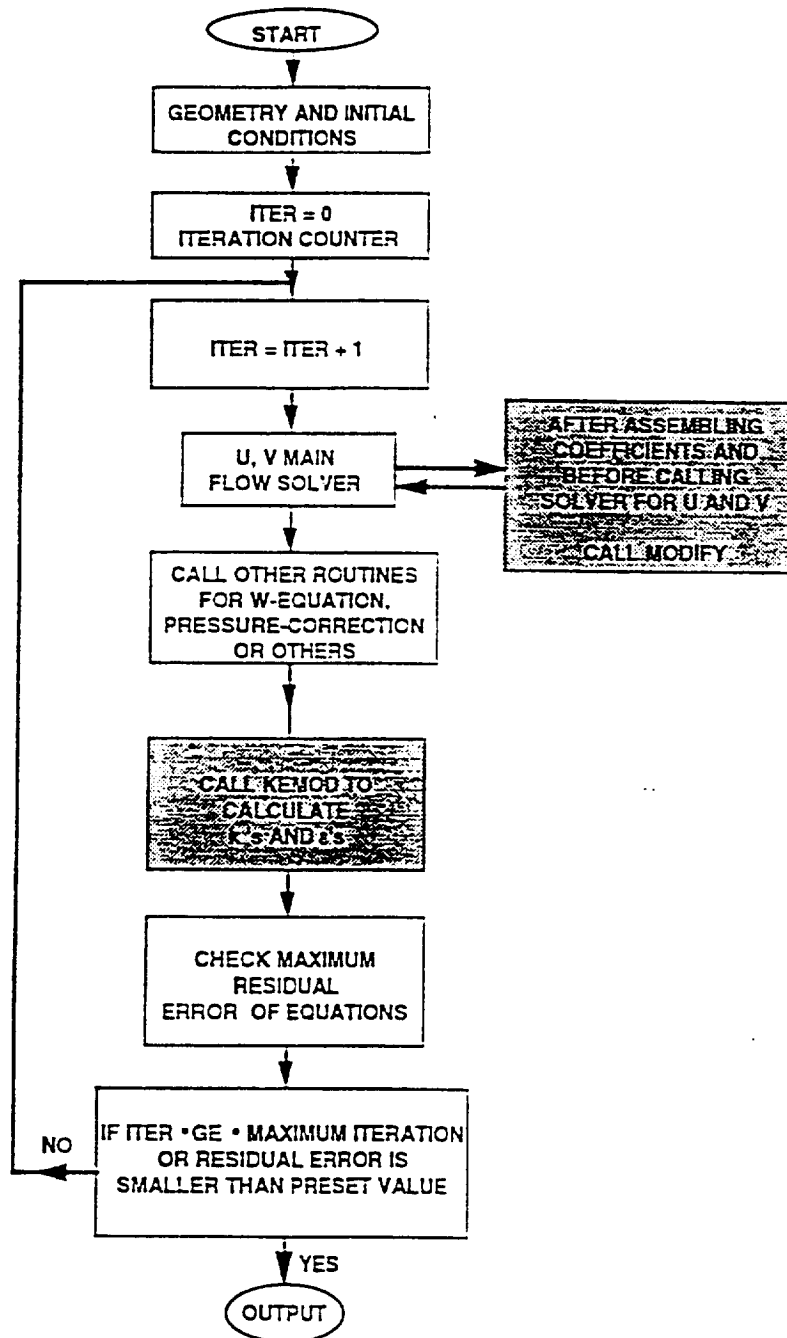
KEMOD-2 FLOW CHART



Rockwell International
Rocketdyne Division

CFD 93 013-017/D1/AHH

KEMOD-2 MODULE DECK (CONT'D)



KEMOD-2 INTERFACE WITH MAIN SOLVER



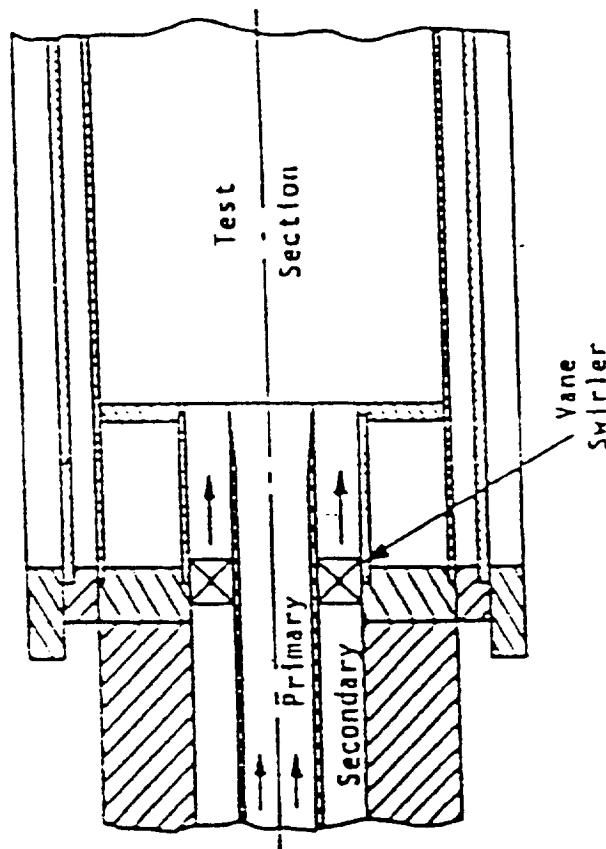
Rockwell International
Rocketdyne Division

CFD 93 013-018/D1/AHH

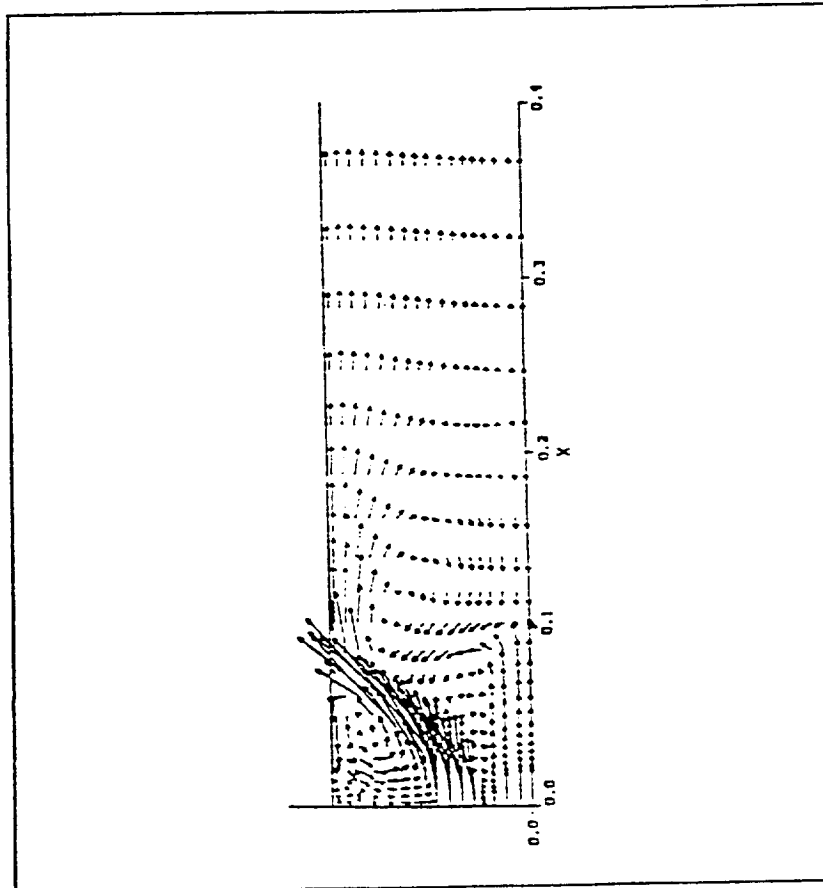
ROBACK AND JOHNSON – SWIRLING COAXIAL JETS DISCHARGING INTO AN EXPANDED DUCT

R. ROBACK AND B. JOHNSON, "MASS AND MOMENTUM TURBULENT
TRANSPORT EXPERIMENT WITH CONFINED SWIRLING COAXIAL JETS,"
NASA CR-168252, 1983

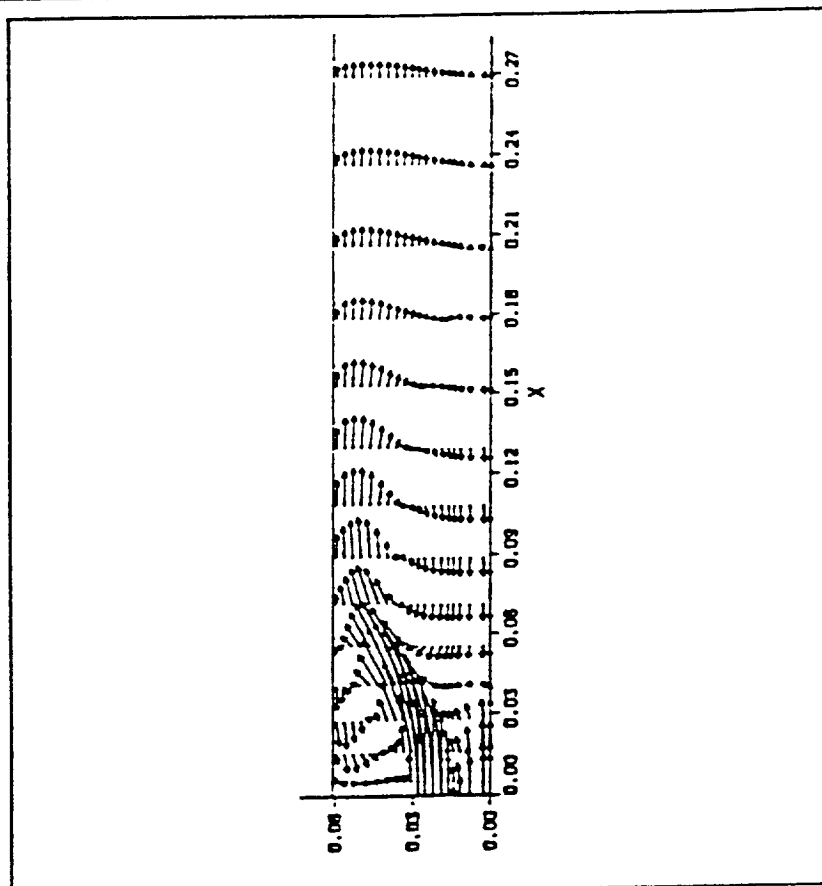
GEOMETRY



ROBACK AND JOHNSON RESULTS VELOCITY VECTORS



SINGLE-SCALE K- ϵ MODEL



MULTI-SCALE K- ϵ MODEL



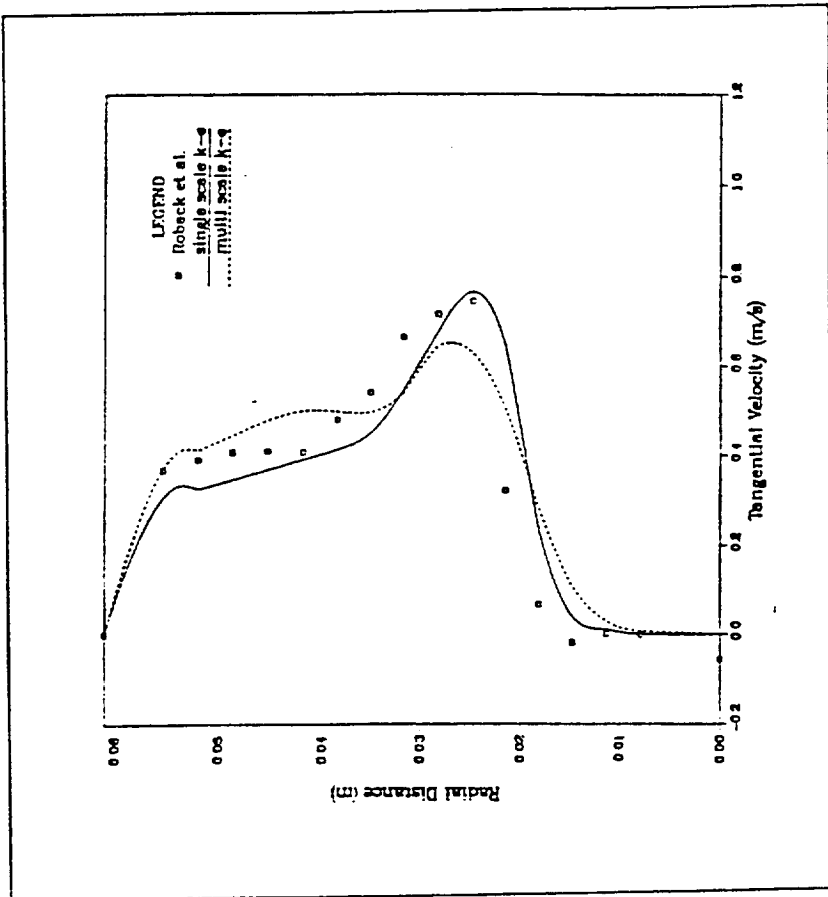
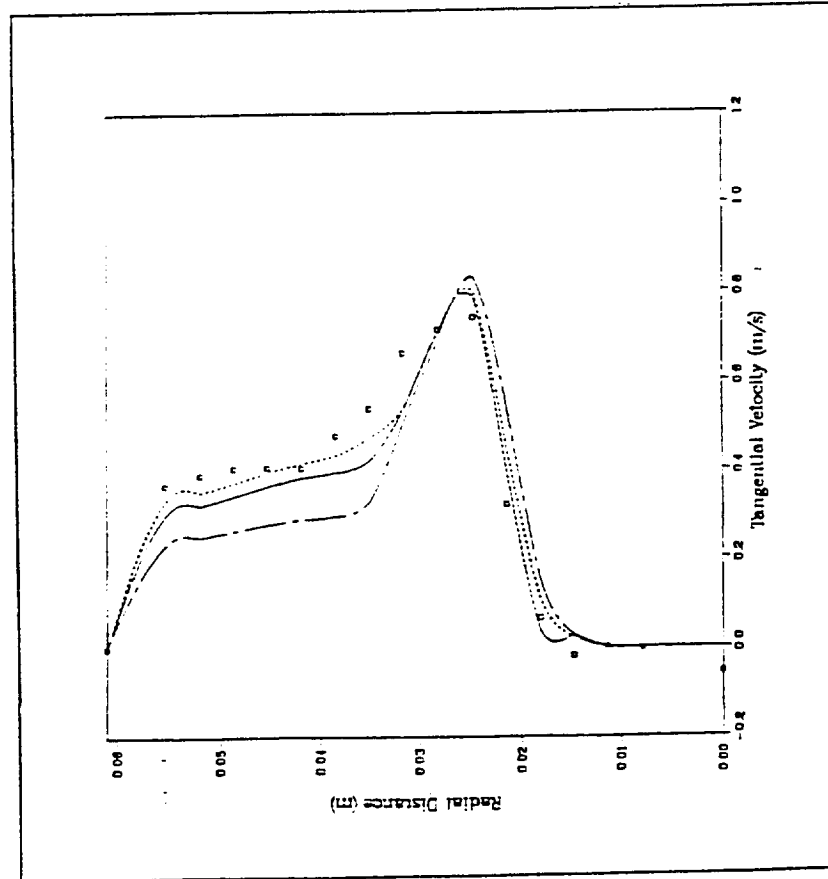
Rockwell International
Rocketdyne Division

CFD 92.032.020/D3/MMS

ROBACK AND JOHNSON RESULTS TANGENTIAL VELOCITY PREDICTIONS AT $X = 0.025 M$

☐ DATA
 — WALL FUNCTION
 - - - LOW-REYNOLDS NO. MODEL
 - . - . 2-LAYER MODEL

☐ DATA
 — SINGLE-SCALE K- ϵ MODEL
 - - - MULTI-TIME SCALE K- ϵ MODEL



SINGLE-SCALE K- ϵ

SUMMARY

- **KEMOD-1 (2-D)**
 - SINGLE SCALE $k-\epsilon$ TURBULENCE MODULE COMPLETE
 - TESTED USING REACT AND USA CODES
- **KEMOD-2 (2-D)**
 - MULTISCALE $k-\epsilon$ TURBULENCE MODULE COMPLETE
 - TESTED USING REACT CODE
- **DEVELOPMENT OF MODULES FOR FULL AND ALGEBRAIC REYNOLDS STRESS MODELS IN PROGRESS**
- **WORK ON 3-D MODULES TO BEGIN AS SCHEDULED (FY '94)**

COMPUTATIONS OF CONFINED SWIRLING FLOWS WITH HIGH ORDER TURBULENCE MODELS IN A MODULAR FORM

A. Hadid¹, M. Sindir¹, C. Chen², and H. Wei²

¹ CFD Technology Center
Rocketdyne Division/Rockwell International
Mail Code IB39, 6633 Canoga Avenue
Canoga Park, CA 91303

² Department of Chemical Engineering
University of Alabama at Huntsville
Huntsville, AL 35899

Abstract

A finite-volume procedure is used to compare the performance of different high order turbulence models for confined swirling turbulent flows. Eddy-viscosity single and multi-scale $k-\epsilon$ turbulence models together with second-moment algebraic and Reynolds stress closure models are tested for a two-dimensional, axisymmetric swirling flow case. The ability of second-moment closure models to capture the interaction between swirl and the turbulent stress field is crucial to the predictive performance of the computational scheme.

To enhance the predictive capability of CFD tools for engineering applications, advanced turbulence models are coded as self-contained module decks that can be interfaced with a number of CFD solvers. Three of these modules, namely the single and the multi-scale models and the Algebraic stress model (ASM) have been successfully interfaced and tested with the code MAST of the University of Alabama at Huntsville in a relatively short time. These modules are independently tested and evaluated with the data of Roback and Johnson for swirling turbulent flow in a confined double concentric jets with a sudden expansion.

Modularization of a general purpose CFD code structure in terms of different aspects of physical models is necessary for computational efficiency. Further, individual modular routines are transportable and can be easily modified to include extra physical effects. This would allow many users using different CFD codes to concentrate their talents on developing and improving physical hypothesis for specific engineering problems.

Introduction

Computational Fluid Dynamics (CFD) has been used extensively for the last decade or so in analyzing complex flow phenomena for many industrial applications, such as, turbomachinery and

combustion devices. Most flows of technological interest are turbulent and for many of them, relatively simple prediction methods are sufficient to produce results of engineering accuracy. For others, mainly in high technology applications, accurate predictions using high order turbulence models are required. Increases in available computational capabilities have permitted the development and testing of sophisticated models in the numerical simulation of turbulent flows. Direct numerical simulation, where all essential scales of the turbulent flow are resolved by solving the unsteady Navier-Stokes equations, are possible only at low to moderate Reynolds numbers. Turbulent flow analysis for engineering applications, therefore, can only be achieved by utilizing the time-averaged Navier-Stokes equations coupled with some level of modelling. The analysis of turbulent transport and modelling evolves from the Reynolds-averaged Navier-Stokes equations and auxiliary equations for velocity and length scales for eddy viscosity specifications towards a more sophisticated modeling strategy - one offering greater width of applicability, particularly in complex shear flows or where external force fields modify the turbulence structure.

One of the widely used models is the two-equation single-time-scale $k-\epsilon$ model⁽¹⁾. In this model transport equations for the turbulence energy (k) and the energy dissipation (ϵ) are solved to determine the turbulent eddy viscosity. An improvement to the single scale $k-\epsilon$ model is the multi-time-scale $k-\epsilon$ model where the energy spectrum of a turbulent flow is split into a production range and a dissipation range⁽²⁾. Improved predictions using the multi-scale over the single-scale $k-\epsilon$ model have been demonstrated^(3,4).

Other complicated single-scale models offering greater width of applicability, particularly in complex shear flows or where external force fields modify the turbulence structure are based on second-moment closures. These take the exact equations for the transport of the Reynolds stresses ($\overline{u_i u_j}$) as their starting point and devise approximations for the

unknown turbulent correlations appearing in them. In a three-dimensional flow, or even in an axisymmetric flow, all six components of the Reynolds stress tensor are nonzero. With a full second-moment closure model (RSM), therefore, differential transport equations need to be solved over the solution domain for each of these components. This represents an increase in the task of numerical solution compared with the situation where the k-ε eddy-viscosity model is adopted. An intermediate level of modeling has evolved^(5,6) known as Algebraic second-moment closure (ASM), with the aim of retaining the greater physical realism of second-moment treatments while achieving computational times closer to that of an eddy-viscosity model. The simplification is achieved by approximating the convective and diffusive transport of the Reynolds stresses in terms of the corresponding transport of turbulent energy. This allows the transport equations for the stresses to be expressed as a set of algebraic formulae containing the turbulence energy and its rate of dissipation as unknowns. Second moment schemes have been extensively and successfully applied to a wide range of flows, as reviewed for example by Leschiziner⁽⁷⁾. Few applications, however, have considered axisymmetric swirling flows^(6,8) where the external forces due to swirl exert damping effects on the turbulent transport.

Progress in turbulence modelling have been paralleled by improvements in numerical techniques, essentially, combining second moment closure with non-orthogonal, co-located grids using finite-volume methods. However, the implementation of RSM into non-orthogonal finite-volume codes poses difficulties: the co-located variable arrangement can cause decoupling of the mean velocity and Reynolds stress fields leading to oscillating solutions or even divergence. Using a special interpolation procedure in the context of Rhie⁽⁹⁾, Obi and Peric⁽¹⁰⁾ calculated the two-dimensional turbulent flow on a co-located grid arrangement using the Reynolds stress turbulence model.

In the present paper, we present predictions of two dimensional/axisymmetric swirling flow using various models based on eddy-viscosity single and multi-scale k-ε and on second moment closure. These models are cast in a modular form enabling them to be used with a number of flow solvers based on the finite-volume and finite-difference methods. A discussion of the different models used and their assessment is presented. The modular structure of the different turbulence models will also be presented and discussed.

Theory and Model Equations

The turbulent flow considered is two-dimensional and steady which can be described by the

Reynolds averaged continuity and momentum equations which may, respectively, be written as

$$\frac{\partial \rho U}{\partial x} + \frac{1}{r} \frac{\partial \rho r V}{\partial r} = 0 \quad (1)$$

$$\frac{\partial \rho r U \Phi}{\partial x} + \frac{\partial \rho r V \Phi}{\partial r} = \frac{\partial}{\partial x} (\eta \mu \frac{\partial \Phi}{\partial x}) + \frac{\partial}{\partial r} (\eta \mu \frac{\partial \Phi}{\partial r}) + r S_{\Phi} \quad (2)$$

Where Φ stands for any of the momentum components U , V , and rW and the corresponding sources S_{Φ} are

$$S_U = -\frac{\partial P}{\partial x} - \frac{\partial \overline{\rho u^2}}{\partial x} - \frac{1}{r} \frac{\partial \overline{\rho r u v}}{\partial r}$$

$$S_V = -\frac{\partial P}{\partial r} + \frac{r W^2}{r} - \frac{2 u V}{r^2} - \frac{1}{r} \frac{\partial \overline{\rho r v^2}}{\partial r} - \frac{\partial \overline{\rho u v}}{\partial x} + \frac{r w^2}{r}$$

$$S_{rW} = -\frac{2 u}{r} \frac{\partial r W}{\partial r} - r \frac{\partial (\overline{\rho u w})}{\partial x} - r \frac{\partial (\overline{\rho v w})}{\partial r} - 2 \overline{\rho v w}$$

where ρ , μ are the fluid density and viscosity respectively.

The appearance of the Reynolds stresses $\overline{u_i u_j}$ represents an unknown correlation and different turbulence models provide the means of relating these unknowns to known determinable quantities.

Single-Scale Eddy-Viscosity Turbulence Models

Here it is assumed that a single-time-scale (proportional to k/ϵ) can be used to describe the turbulent flow. Turbulence is simulated through transport equations for the turbulent kinetic energy (k), and its rate of dissipation (ϵ). The stress tensor is modelled using a gradient transport model of the form

$$-\overline{u_i u_j} = \nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij} \quad (3)$$

The generalized form of the two-equation eddy-viscosity turbulence model can be written as Kinetic Energy (k) equation:

$$C_k = D_k + P - \epsilon \quad (4)$$

where

$$C_k = \frac{\partial U_j k}{\partial x_j} \quad \text{Convection of } k$$

$$D_k = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] \quad \text{Diffusion of } k$$

$$P = -\overline{u_i u_j} \frac{\partial U_i}{\partial x_j} = -\nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \frac{\partial U_i}{\partial x_j} \quad \text{Production of } k$$

$$\nu_t = \text{eddy viscosity} = C_\mu \frac{k^2}{\varepsilon}$$

Energy Dissipation (ε) equation:

$$C_\varepsilon = D_\varepsilon + \frac{\varepsilon}{k} (C_{\varepsilon 1} P - C_{\varepsilon 2} \varepsilon) \quad (5)$$

where

$$C_\varepsilon = \frac{\partial U_j \varepsilon}{\partial x_j} \quad \text{Convection of } \varepsilon$$

$$D_\varepsilon = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] \quad \text{Diffusion of } \varepsilon$$

In the present study, the standard two-equation model was used with the wall function⁽¹⁾ and the two-layer model⁽¹¹⁾ to bridge the gap between the near-wall log-layer region and the fully turbulent region away from the wall. In the standard model the numerical values of the constants are $C_\mu=0.09$, $C_{\varepsilon 1}=1.44$, $C_{\varepsilon 2}=1.92$, $\sigma_k=1.0$ and $\sigma_\varepsilon=1.3$. Details of the implementation of the wall function and the two-layer models can be found in Hadid and Sindir⁽¹²⁾.

Multi-Time-Scale k- ε Turbulence Model

The Multi-time-scale turbulence model used here is based on the variable energy partitioning of the turbulent energy spectrum proposed by Kim and Chen⁽³⁾. In this model the turbulent kinetic energy spectrum is divided into two sets of wave number regions giving two evolution equations for each region. These equations represent the kinetic energy (k_p) and the energy dissipation (ε_p) in the production range of the spectrum and the kinetic energy (k_t) and the energy dissipation (ε_t) in the dissipation range of the spectrum. This model allows the partition to move toward the high wave number region when production is high and toward the low wave number region when production vanishes.

The equations for the turbulent kinetic energy (k_p) and the energy transfer rate (ε_p) for the production range are

$$C_{k_p} = D_{k_p} + P - \varepsilon_p \quad (6)$$

$$C_{\varepsilon_p} = D_{\varepsilon_p} + \frac{P}{\rho k_p} \left(\frac{1}{\rho} C_{p1} P + C_{p2} \varepsilon_p \right) - C_{p3} \frac{\varepsilon_p^2}{k_p} \quad (7)$$

The equations for the turbulent kinetic energy (k_t) and the dissipation rate (ε_t) for the high wave number transfer region are

$$C_{k_t} = D_{k_t} + \varepsilon_p - \varepsilon_t \quad (8)$$

$$C_{\varepsilon_t} = D_{\varepsilon_t} + \frac{\varepsilon_p}{k_t} (C_{t1} \varepsilon_p + C_{t2} \varepsilon_t) - C_{t3} \frac{\varepsilon_t^2}{k_t} \quad (9)$$

$$\text{where } C_{k_p} = \rho \frac{\partial U_j k_p}{\partial x_j} \quad \text{and} \quad C_{\varepsilon_p} = \rho \frac{\partial U_j \varepsilon_p}{\partial x_j}$$

$$D_{k_p} = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_{k_p}} \right) \frac{\partial k_p}{\partial x_j} \right] \quad \text{and}$$

$$D_{\varepsilon_p} = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_{\varepsilon_p}} \right) \frac{\partial \varepsilon_p}{\partial x_j} \right]$$

similarly for C_{k_t} , C_{ε_t} , D_{k_t} , and D_{ε_t} equations and the model constants used are those of Kim and Chen⁽³⁾.

The terms $\left(\frac{1}{\rho} C_{p1} \frac{P^2}{k_p} \right)$ and $\left(\rho C_{t1} \frac{\varepsilon_p^2}{k_t} \right)$ represent variable energy transfer functions. The former increases the energy transfer rate when production is high and the latter increases the dissipation rate when the energy transfer rate is high. The turbulent viscosity μ_t is given as $\mu_t = \rho C_\mu k^2 / \varepsilon_p = \rho C_\mu k^2 / \varepsilon_t$, where $k = k_p + k_t$ is the total turbulent kinetic energy.

Second Moment Closure Models

The exact form of the Reynolds stress equations can be derived from the time-averaged form of the Navier-Stokes equations and can be written as:

$$\frac{D}{Dt} (\overline{\rho u_i u_j}) = P_{ij} + \Phi_{ij} + D_{ij} + \rho \varepsilon_{ij} \quad (10)$$

where

$$P_{ij} = -\rho \left(\overline{u_j u_k} \frac{\partial u_i}{\partial x_k} + \overline{u_i u_k} \frac{\partial u_j}{\partial x_k} \right) \quad \text{Production}$$

$$D_{ij} = \frac{\partial}{\partial x_k} \left(-\rho \overline{u_i u_j u_k} - \delta_{ik} \overline{p u_j} - \delta_{jk} \overline{p u_i} \right) \quad \text{Diffusion}$$

$$\Phi_{ij} = \rho \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad \text{Pressure-strain redistribution}$$

$$\varepsilon_{ij} = \frac{2}{3} \delta_{ij} \varepsilon \quad \text{Dissipation}$$

Due to the introduction of correlations of higher orders, modelling of these terms is required to close the set of equations.

Algebraic Stress Model (ASM)

The ASM model used is based on the work of Rodi⁽⁵⁾. The idea is to simplify the stress equation (eq. 10) by approximating the convective and diffusive

transport of the Reynolds stresses ($\overline{u_i u_j}$) in terms of the corresponding transport of turbulent energy. This simplification allows the transport equation of the stresses to be expressed as a set of algebraic formulae containing the turbulent energy and its rate of dissipation as unknowns. This set of algebraic equations can be written as;

$$\overline{u_i u_j} = \frac{k}{P-\epsilon} [P_{ij} - \frac{2}{3}\delta_{ij}\epsilon + \Phi_{ij}] \quad (11)$$

The pressure-strain term Φ_{ij} is decomposed into a fluctuating part ($\Phi_{ij,1}$), a part due to the mean rate of strain ($\Phi_{ij,2}$), and a part due to reflected wall-influence ($\Phi_{ij,w}$), i.e., $\Phi_{ij} = \Phi_{ij,1} + \Phi_{ij,2} + \Phi_{ij,w}$

Rotta's return to isotropy concept is used to model the non-linear part ($\Phi_{ij,1}$) as

$$\Phi_{ij,1} = -C_1 \frac{\epsilon}{k} (\overline{u_i u_j} - \frac{2}{3}k\delta_{ij})$$

$\Phi_{ij,2}$ is modelled using the isotopization of production concept as

$$\Phi_{ij,2} = -C_2 (P_{ij} - \frac{2}{3}P\delta_{ij})$$

The wall reflection term $\Phi_{ij,w}$ is modelled following Shih⁽¹³⁾ and Gibson and Launder⁽¹⁴⁾ as

$$\Phi_{ij,w} = \Phi_{ij,1w} + \Phi_{ij,2w}$$

where

$$\Phi_{ij,1w} =$$

$$C_1 \rho \frac{\epsilon}{k} (\overline{u_k u_m} n_k n_m \delta_{ij} - \frac{3}{2} \overline{u_k u_i} n_k n_j - \frac{3}{2} \overline{u_k u_j} n_k n_i) f \quad (12)$$

$$\Phi_{ij,2w} =$$

$$C_2 (\Phi_{km,2} n_k n_m \delta_{ij} - \frac{3}{2} \Phi_{ik,2} n_k n_j - \frac{3}{2} \Phi_{jk,2} n_k n_i) f \quad (13)$$

where n_i is the wall-normal unit vector in the i -direction. The wall distance function (f) represents the ratio of turbulence length scale and the wall distance

$$f = \left(\frac{C_m^{0.75} k^{1.5}}{\kappa \epsilon} \right) \frac{1}{\Delta n} \text{ where } \Delta n \text{ is the wall-normal distance. The above wall-correction terms are written in a tensorially invariant form and their effect is to transfer energy from the wall-normal normal stress component to the tangential stresses i.e it is redistributive.}$$

For axisymmetric swirling flows the set of algebraic stress equations can be written in a general matrix form as $\underline{A} \underline{T} = \underline{B}$ where

$$\underline{A} =$$

$$\begin{array}{cccccc} \frac{3\epsilon}{2\lambda k} + 2\frac{\partial U}{\partial x} & \frac{\partial V}{\partial y} & \frac{V}{r} & 2\frac{\partial U}{\partial y} - \frac{\partial V}{\partial x} & \frac{\partial W}{\partial y} + \frac{W}{r} & \frac{\partial W}{\partial x} \\ \frac{\partial U}{\partial x} & \frac{3\epsilon}{2\lambda k} + 2\frac{\partial V}{\partial y} & \frac{V}{r} & 2\frac{\partial V}{\partial x} - \frac{\partial U}{\partial y} & -(\frac{\partial W}{\partial y} + 2\frac{W}{r}) & \frac{\partial W}{\partial x} \\ \frac{\partial U}{\partial x} & \frac{\partial V}{\partial y} & \frac{3\epsilon}{2\lambda k} + 2\frac{V}{r} & -(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}) & 2\frac{\partial W}{\partial y} + \frac{W}{r} & 2\frac{\partial W}{\partial x} \\ \frac{\partial V}{\partial x} & \frac{\partial U}{\partial y} & 0 & \frac{\epsilon}{\lambda k} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} & 0 & \frac{W}{r} \\ 0 & \frac{\partial W}{\partial y} & \frac{W}{r} & \frac{\partial W}{\partial x} & \frac{\epsilon}{\lambda k} + \frac{\partial V}{\partial y} + \frac{V}{r} & \frac{\partial V}{\partial x} \\ \frac{\partial W}{\partial x} & 0 & 0 & \frac{\partial W}{\partial y} & \frac{\partial U}{\partial y} & \frac{\epsilon}{\lambda k} + \frac{\partial U}{\partial x} + \frac{V}{r} \end{array}$$

$$\underline{T} = [\rho \overline{u u}, \rho \overline{v v}, \rho \overline{w w}, \rho \overline{u v}, \rho \overline{v w}, \rho \overline{u w}]^T$$

$$\underline{B} = \begin{array}{l} \frac{\rho \epsilon}{\lambda} + \frac{3}{2(1-C_2)} (\Phi_{11,1w} + \Phi_{11,2w}) \\ \frac{\rho \epsilon}{\lambda} + \frac{3}{2(1-C_2)} (\Phi_{22,1w} + \Phi_{22,2w}) \\ \frac{\rho \epsilon}{\lambda} + \frac{3}{2(1-C_2)} (\Phi_{33,1w} + \Phi_{33,2w}) \\ \frac{3}{2(1-C_2)} (\Phi_{12,1w} + \Phi_{12,2w}) \\ \frac{3}{2(1-C_2)} (\Phi_{23,1w} + \Phi_{23,2w}) \\ \frac{3}{2(1-C_2)} (\Phi_{13,1w} + \Phi_{13,2w}) \end{array}$$

$$\text{and } \lambda = \frac{1-C_2}{C_1 - 1 + \frac{P}{\rho \epsilon}}$$

Reynolds Stress Model (RSM)

In the RSM model the full transport equation for the Reynolds stresses (eq.10) are solved for each stress

component ($\overline{u_i u_j}$) after modelling the diffusion and the pressure strain terms similar to Launder et. al⁽¹⁵⁾. The diffusion term is modelled as

$$D_{ij} = -\frac{\partial}{\partial x_k} [\rho C_k \overline{u_k u_i} \frac{k}{\epsilon} \frac{\partial \overline{u_k u_j}}{\partial x_k}]$$

The pressure-strain redistribution term Φ_{ij} is modelled in a similar way to that used in the ASM model discussed earlier. Special consideration is given to the problem of mean velocity-Reynolds stress

decoupling which appear when using a collocated grid arrangement which is a source of numerical instability. This is done by invoking a special interpolation procedure for the cell-face stresses in the context of Rhie⁽⁹⁾. This practice results in the addition of normal stresses to the pressure term where the cell-face velocity is sensitized to the pressure differences as well as to normal stress differences at the nodes surrounding the face.

Turbulence Model Decks (Modules)

As the state-of-the-art of computers has advanced, so has the range, size and complexity of flow models being applied. Users have become more sophisticated and there is a constant demand for improvement. CFD codes have adapted to this demand and many general-purpose computer codes have been developed and used. As general-purpose codes become larger, their code structure becomes sophisticated. In general codes can be divided into three main areas, they include; 1) Numerical algorithms (which can be subdivided into discretization methods and solution techniques). 2) Methods of dealing with complex geometries. 3) Physical models (which include turbulence models, porosity, combustion kinetics, two-phase flow...). It seems, therefore, that the practicing engineer must have the knowledge of all these elements of the CFD program in order to successfully utilize this code. To obtain the maximum benefits from these general-purpose CFD codes, modularization of the code structure may be necessary. That is developing individual modular routines for the solver and for different physical models for example. If such modules are successful it would allow users to concentrate their talents on developing and improving physical hypothesis such as turbulent models for example that can easily be tested using such modules.

In the present work, turbulent modules are being developed to meet this need. Figure 1, shows a flowchart of a turbulence module interfaced with a typical main flow solver. The module is called by the flow solver passing to it the mean flow velocities, mass fluxes at cell faces and grid information among others. The turbulence differential equations are discretized and the matrix coefficients are setup and solved using Stones strongly implicit method⁽¹⁶⁾. In the ASM module, the set of algebraic stress equations are solved simultaneously using Gauss-Seidel method at each step or iteration. In the eddy-viscosity models the values of k , ϵ , and eddy viscosity (μ_t) are passed to the main flow solver, while, in the second moment

closure models the Reynolds stresses $\overline{u_i u_j}$ are passed to the main solver. The solver then calls subroutine MODIFY of the module where the momentum

sources are modified to account for the near-wall shear stresses in the eddy-viscosity models or to calculate Reynolds stress gradients in the second moment models.

These modules are structured to be self-contained and transportable to a number of general purpose CFD solvers to maximize computational efficiency. They have been tested independently at the University of Alabama at Huntsville using the MAST code.

Results

The various turbulence models are analyzed by comparing model predictions with the experimental data of Roback and Johnson⁽¹⁷⁾ for swirling flow in confined double concentric jets with a sudden expansion.

Figure 2, shows the decay of the mean axial centerline velocity using both the single and multi-scale $k-\epsilon$ models. Figure 2a, shows the comparison using the wall-function near wall approach and Figure 2b shows the results using the two-layer near wall model. The single-scale $k-\epsilon$ model seems to underpredict the extent of the central recirculation zone as compared with the multi-scale $k-\epsilon$ model. Moreover, improved comparisons with the data are obtained using the two-layer near wall model. Figure 3, shows the radial profile of the mean axial velocity at two distances downstream of the jet exit. Again, the two-layer model predicts better comparisons with the data than the wall function approach. The radial profiles of the mean tangential velocity are shown in Figure 4.

Figure 5, shows the radial profile of the mean axial velocity at three axial locations using the algebraic stress model (ASM) with wall function and two-layer near wall models. The radial profiles of the rms axial turbulent intensity are shown in Figure 6. Streamline contours are shown in Figure 7 using the single-scale $k-\epsilon$ and the ASM models with the two-layer near wall approach. The extent of the central vortex is better predicted using the ASM model. Preliminary results were also obtained using the full Reynolds stress model (RSM). Comparisons with the backward facing step data of Driver and Seegmiller⁽¹⁸⁾ shows improved predictions over the single scale $k-\epsilon$ model as shown in Figure 8 where the radial profiles of the axial normal stress and shear stress are plotted at four step heights downstream. Further testing of the RSM model for swirling flows are planned.

Conclusions

Different turbulence models for industrial applications have been formatted in a modular form and successfully interfaced and tested independently using two different main flow solvers. The turbulence models include the single and multi-scale $k-\epsilon$ models both with wall functions and two-layer near wall models. Second moment models that include the algebraic (ASM) and full Reynolds stress model (RSM) have been tested. It was shown that the two-layer near wall model improves predictions as compared to the wall function approach. Convergence of the stiff ASM model equations was obtained by solving the 6×6 stress equations (for axisymmetric/swirling flows) at each iteration. The wall-reflection terms in the pressure-strain model showed little or no improvements in the ASM model predictions. Elaborate pressure-strain models that require no wall-damping are needed e.g. Speziale et.al⁽¹⁹⁾. The full Reynolds stress model (RSM) promises to be the next model to be used for engineering applications.

References

- (1) Launder B., and Spalding D., Comp. Meth. in Appl. Mech. & Eng. vol. 3, 1974, pp. 269-289
- (2) Hanjalic K., Launder B., and Schiestel R., Turb. Shear Flows, eds. Bradbury L. et. al, vol.2, Springer-Verlag, N.Y. 1980, pp. 36-49
- (3) Kim S.-W and Chen C.-P, Num. Heat Transfer, pt B, vol. 16, no.2, 1989, pp.193-211
- (4) Kim S.-W, Num. Heat Transfer, pt B., vol. 17, 1990, pp.101-122
- (5) Rodi W., Z. Ang. Math. und Mech., vol 56, 1976, pp.219-
- (6) Fu S., Huang P., Launder B., and Leschziner M., ASME J. Fluid Eng., vol.110, 1988, pp.216-221
- (7) Leschziner M., Proc. of Int. Form on Math. Modeling of Processes in Energy Systems, Int. Center for Heat and Mass Transfer, Sarajevo, 1989
- (8) Hogg S., and Leschziner M., AIAA J., vol.27, no.1, 1989, pp.57-63
- (9) Rhie C., Univ. of Ill., Urbana-Champaign, Ph.D. Thesis, 1981
- (10) Obi S., and Peri'c M., AIAA J., vol.29, no.4, 1991, pp.585-590
- (11) Chen H., and Patel V., AIAA J., vol. 26, no. 6, 1988, pp.641-648
- (12) Hadid H., and Sindir M., NASA CP-3174, (1992)
- (13) Shir C., J. Atmos. Sci., vol.30, 1973, pp.1327-
- (14) Gibson M., and Launder B., J. Fluid Mech., vol.86, pt.3, 1978, pp.471-
- (15) Launder B., Reece G., and Rodi W., J. Fluid

Mech., vol.68, 1975, pp.537-566

- (16) Stone H., SIAM J. num. anal. vol.5, 1968, pp.530-

- (17) Roback R., and Johnson B., NASA-CR 168252 (1983)

- (18) Driver D., and Seegmiller H., AIAA Paper 82-1029, (1982)

- (19) Speziale C, Sarkar S., and Gatski T., J. Fluid Mech., vol.227, 1991, pp.245-272

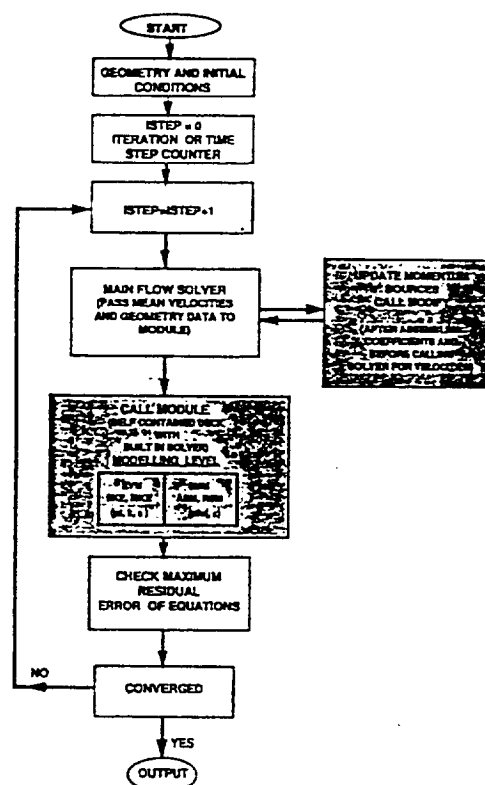


Figure 1. Typical main flow solver interfaced with a turbulence module

CONFINED SWIRLING JET FLOW (ROBACK & JOHNSON)
Decay of mean axial centerline velocity

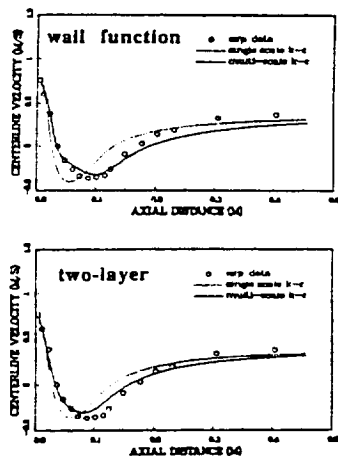


Figure 2. Mean axial centerline velocity
----- single-scale $k-\epsilon$ model
—— multi-scale $k-\epsilon$ model

CONFINED SWIRLING JET FLOW (ROBACK & JOHNSON)
—— STANDARD $k-\epsilon$, —— MULTI-SCALE $k-\epsilon$

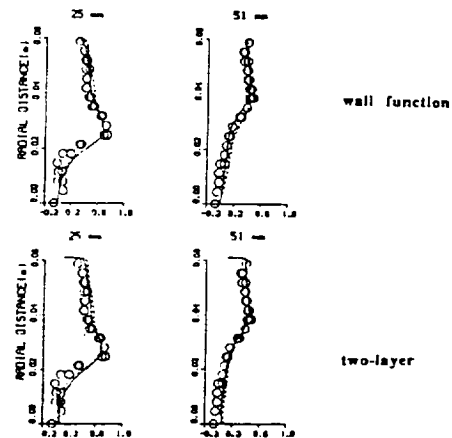


Figure 4. Radial profiles of mean tangential velocity

CONFINED SWIRLING JET FLOW (ROBACK & JOHNSON)
—— STANDARD $k-\epsilon$, —— MULTI-SCALE $k-\epsilon$

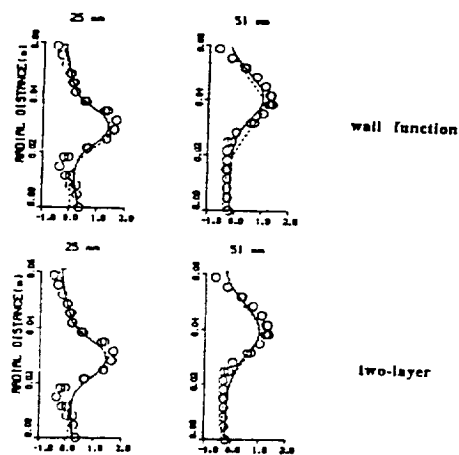


Figure 3. Radial profile of mean axial velocity

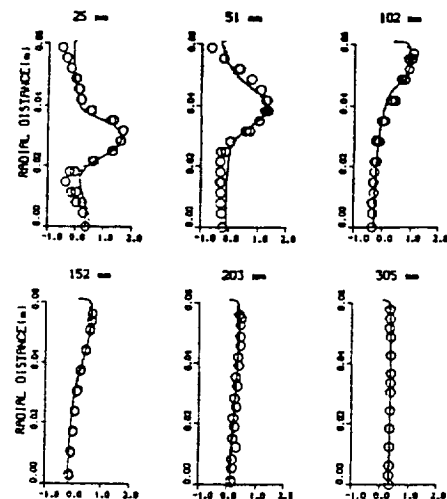


Figure 5. Radial profiles of mean axial velocity
ASM turbulence model
----- wall function model
—— two-layer model

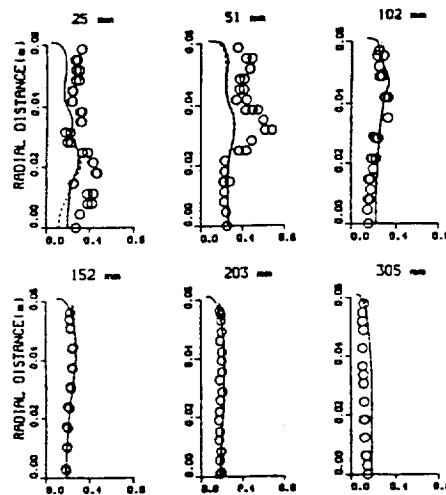


Figure 6. Radial profiles of the axial turbulent intensity
 ASM turbulence model
 --- wall function model
 — two-layer model

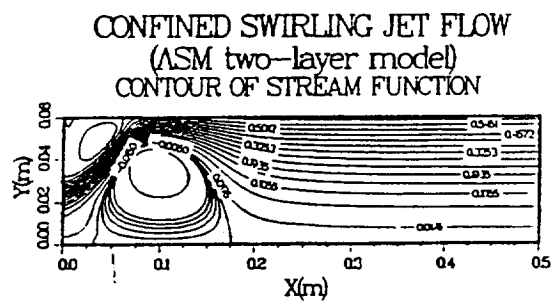
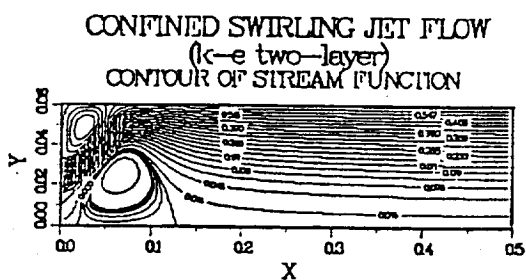


Figure 7. Streamline contours

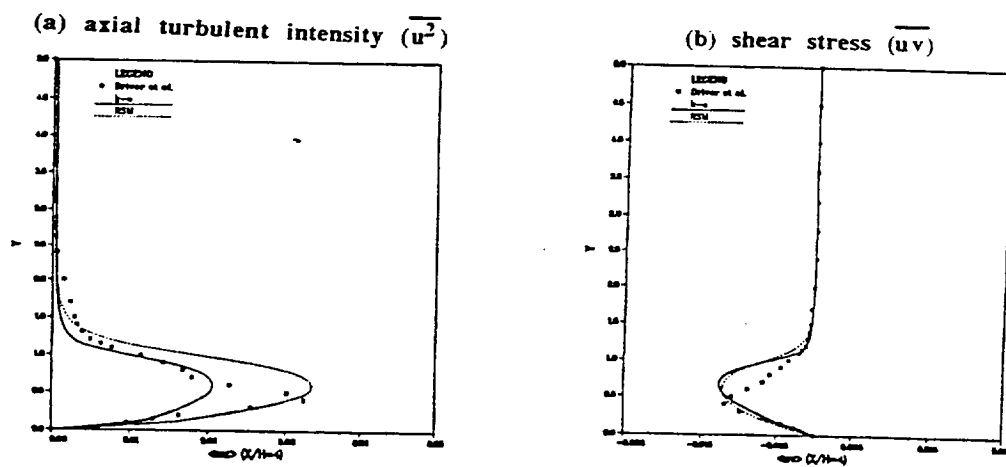


Figure 8. Backward facing step (Driver & Seegmiller)
 — single-scale $k-\epsilon$ model
 --- RSM model

A NUMERICAL STUDY OF TWO-DIMENSIONAL VORTEX SHEDDING FROM RECTANGULAR CYLINDERS

A. H. HADID[†] M. M. SINDIR[†] R. I. ISSA[‡]

Abstract

An efficient time-marching, noniterative calculation method is used to analyze time-dependent flows around rectangular cylinders. The turbulent flow in the wake region of a square section cylinder is analyzed using an anisotropic $k-\epsilon$ model. Initiation and subsequent development of the vortex shedding phenomenon is naturally captured once a perturbation is introduced in the flow. Transient calculations using standard eddy-viscosity and anisotropic $k-\epsilon$ models, averaged over an integral number of cycles to get the fluctuating energy (organized and turbulent), are compared with experimental data. It is shown that the anisotropic $k-\epsilon$ model resolves the anisotropy of the Reynolds stresses and gives mean energy distribution closer to the experiment than the standard $k-\epsilon$ model.

1. INTRODUCTION

Vortex shedding is a periodic unsteady flow phenomenon that occurs frequently behind bluff bodies and is therefore of great practical importance. Many attempts to calculate the two-dimensional (2-D) vortex shedding motion past square and circular cylinders by solving the unsteady Navier-Stokes equations were successful at low Reynolds numbers where the flow is laminar and the fluctuations are periodic, e.g., [1] and [2]. At higher Reynolds numbers which are more relevant in practice, turbulent fluctuations are superimposed on the periodic unsteady motion. The problem then concerns the decomposition of the flow into organized motion that is resolved in the calculation and a remaining turbulent motion to be represented by a turbulence model. Previous analysis of vortex shedding calculations at high Reynolds numbers have not been successful due to the inadequacy of the standard $k-\epsilon$ model and the lack of affordable higher order models that take into account the anisotropy of the turbulent intensities

Franke *et al.* [3] analyzed the unsteady turbulent flow for a square cylinder using the standard $k-\epsilon$ model. They showed that the model tends to damp the periodic shedding motion underpredicting the Strouhal number. They also analyzed the detailed experimental results of Cantwell and Coles [4] for vortex shedding in the 2-D wake behind a circular cylinder. They additionally point out the need for improved models that account for the history and transport effects of the individual stresses. MacInnes *et al.* [5] used the standard $k-\epsilon$ model to simulate the periodically forced turbulent mixing layer investigated experimentally by Weisbrodt and Wygnanski [6]. They managed to capture the main features of the mixing layer development where there is a clear distinction between the organized and the random turbulent motion.

Presented at the 4th International Symposium on Computational Fluid Dynamics - Davis on September 9 - 12, 1991. Received on March 21, 1992. Received in final form on May 12, 1992.

[†] CFD Technology Center, Rockwell Int./Rocketdyne Div., 6633 Canoga Ave., Canoga Pk., CA 91303, USA

[‡] Dept. Mineral Resources Engineering, Imperial College of Science, Technology and Medicine, London, SW7, 2BP, England

Majumdar and Rodi [7] have shown that the separated turbulent flow past circular cylinders cannot be predicted realistically without a time-accurate numerical procedure to account for the periodic shedding of vortices.

Experimental investigations are needed to judge the different numerical and turbulent schemes. Durão *et al.* [8] conducted an experimental study of transient turbulent flow behind a square cylinder. They used spectral analysis and digital filtering of the LDV data in order to separate and quantify the turbulent and periodic, nonturbulent motions. They show for example that in the zone of highest velocity fluctuations the energy associated with the turbulent fluctuations is about 40 % of the total energy. Therefore, for a successful simulation of transient turbulent flows, a reliable time-accurate numerical procedure and a good turbulence model are needed.

The purpose of the present paper is to model turbulent vortex shedding flows using an efficient time-accurate numerical procedure based on the PISO [9] methodology. Calculations of the turbulent vortex shedding are performed using the two-equation $k-\varepsilon$ model with isotropic eddy-viscosity and with a modified two-equation model using an anisotropic eddy-viscosity. In the anisotropic model, nonlinear corrections are added to improve the eddy-viscosity representation of the Reynolds stresses as developed by Yoshizawa [10] with the aid of a two-scale direct interaction approximation. A similar anisotropic eddy-viscosity model was also developed by Speziale [11]. The adequacy of the models to simulate transient turbulent flows is assessed with the aid of the experimental results of Durão *et al.* [8] for vortex shedding in the 2-D wake behind a square cylinder at $Re = 14,000$.

2. MODEL EQUATIONS

The basic equations of motion in transient periodic flows can be written after separating the flow into an organized (phase averaged) component

$$U_i(x_i, t) = \frac{1}{N} \sum_{n=0}^N u_i(x_i, t + nT) \quad (1)$$

where $U_i(x_i, t)$ is the resolvable portion of the instantaneous velocity u_i , and T is the period of the oscillation, and a random turbulent component $u'_i(x_i, t)$. The instantaneous velocity $u_i(x_i, t)$ is then given by

$$u_i = U_i + u'_i = \bar{u}_i + \tilde{u}_i + u'_i \quad (2)$$

where \bar{u}_i is the time-mean component of the velocity, and \tilde{u}_i is the periodic fluctuating component. Assuming an incompressible flow, the momentum equations can be written after applying phase averaging as;

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\nu \frac{\partial U_i}{\partial x_j} + R_{ij} \right) \quad (3)$$

where $R_{ij} = -\langle u'_i u'_j \rangle$ is the phase-averaged Reynolds stress tensor and ν is the kinematic viscosity.

Standard Isotropic $k-\varepsilon$ Model

In the standard isotropic $k-\varepsilon$ model [12], R_{ij} is approximated by using the eddy-viscosity ν_t as;

$$R_{ij} = -\frac{2}{3} k \delta_{ij} + \nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \quad (4)$$

where k is the phase-averaged turbulent kinetic energy and $\nu_t = C_\mu(k^2/\epsilon)$, ϵ is the phase-averaged energy dissipation rate and C_μ is a model constant. The spatial and temporal distribution of k and ϵ are determined from differential transport equations of these quantities

$$\frac{\partial k}{\partial t} + U_i \frac{\partial k}{\partial x_i} = \frac{\partial}{\partial x_i} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_i} \right] + G - \epsilon \quad (5)$$

$$\frac{\partial \epsilon}{\partial t} + U_i \frac{\partial \epsilon}{\partial x_i} = \frac{\partial}{\partial x_i} \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_i} \right] + \frac{\epsilon}{k} (C_1 G - C_2 \epsilon) \quad (6)$$

where $G = R_{ij} \frac{\partial U_i}{\partial x_j}$ is the turbulent generation term. The constants C_μ , C_1 , C_2 , σ_k , and σ_ϵ have values of 0.09, 1.44, 1.92, 1.0, and 1.3, respectively.

Anisotropic k- ϵ Model

In the anisotropic model the Reynolds stresses can be expressed as;

$$R_{ij} = -\frac{2}{3}k\delta_{ij} + \nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) + \frac{1}{3} \left(\sum_{m=1}^3 \tau_m S m_{kk} \right) \delta_{ij} - \sum_{m=1}^3 \tau_m S m_{ij} \quad (7)$$

$$\tau_m = C\tau_m \frac{k^3}{\epsilon^2} \quad (8)$$

$$S1_{ij} = \frac{\partial U_i}{\partial x_k} \frac{\partial U_j}{\partial x_k} \quad (9)$$

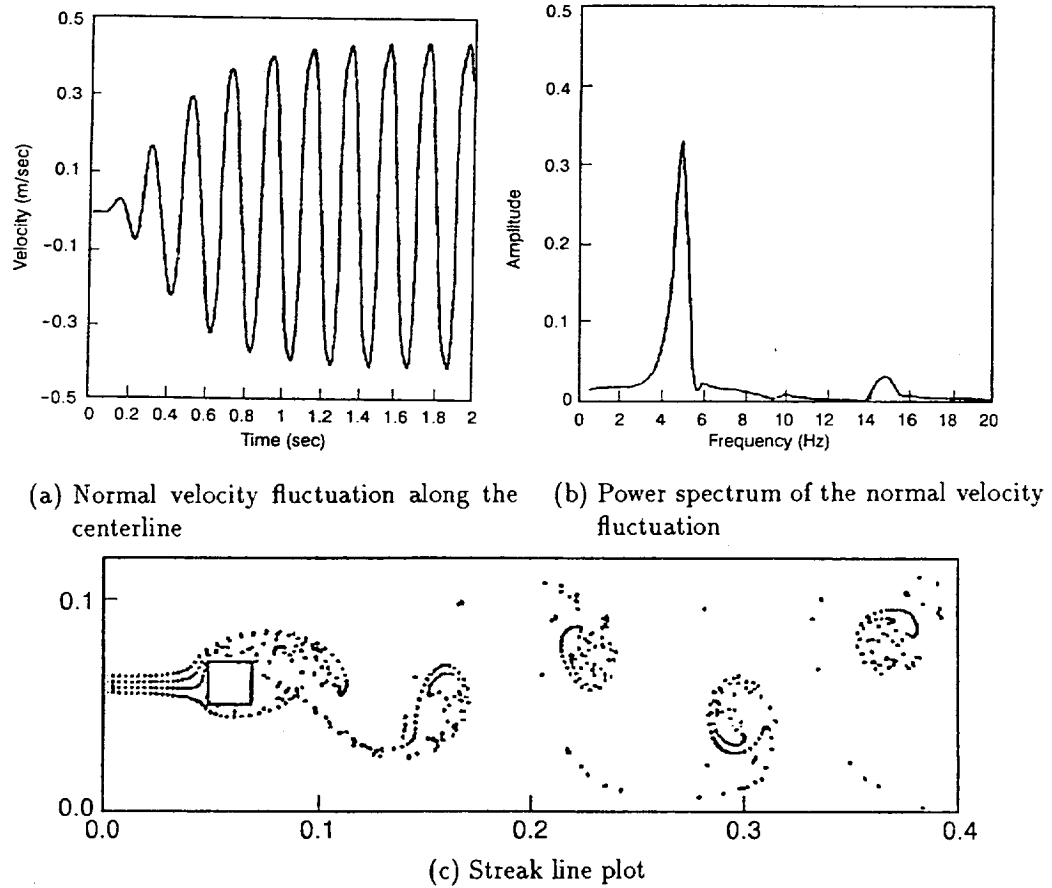
$$S2_{ij} = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_k} \frac{\partial U_k}{\partial x_j} + \frac{\partial U_j}{\partial x_k} \frac{\partial U_k}{\partial x_i} \right) \quad (10)$$

$$S3_{ij} = \frac{\partial U_k}{\partial x_i} \frac{\partial U_k}{\partial x_j} \quad (11)$$

and $C\tau_m$ ($m = 1, 2, 3$) are model constants. The first two terms on the right hand side of (7) give the familiar isotropic eddy-viscosity representation, while the third and fourth terms express the anisotropy of R_{ij} . These additional nonlinear quadratic terms of the mean velocity gradients seem to be a simple way to resolve the individual normal stresses with the k- ϵ model. The anisotropy is reflected especially in the k- ϵ equation where both the diffusion and production terms are quadratic forms of the mean velocity gradients and turbulent kinetic energy gradients.

The anisotropic eddy-viscosity model has been successfully used by Nisizima and Yoshizawa [13] and Myong and Kasagi [14] for fully developed turbulent channel flows. In their calculations only $C\tau_1$ and $C\tau_2$ were optimized to reproduce the anisotropy of the turbulent intensities since $C\tau_3$ does not appear in their equations. In the present study the flow is shear dominated with little departure from isotropy. Therefore, the model constants $C\tau_1$, $C\tau_2$, and $C\tau_3$ were optimized to 0.01, 0.01, and 0.001, respectively, to satisfy the realizability constraint. (Note: zero constants reduce to the isotropic eddy-viscosity model.)

Applications of the k- ϵ isotropic and anisotropic eddy-viscosity models were made using wall functions to bridge the viscosity affected near the obstacle wall region. It is assumed that inadequacies in near-wall modelling play a minor role to the inaccuracy of normal Reynolds

Fig.1: Flow characteristics of the wake for $Re = 14,000$

stress differences arising from use of an isotropic eddy viscosity. Improvements can be made by integrating all the way to the wall [15] or by using the two-layer model of Chen and Patel [16].

3. NUMERICAL METHOD

The PISO methodology [9], in conjunction with a finite-volume technique, is used to solve the implicitly discretized, time-dependent flow equations. The method is essentially noniterative, where the solution process is split into a series of steps whereby operations on pressure are decoupled from those on velocity at each time-step. The avoidance of iterations substantially reduces the computational effort compared with that required by iterative methods. This is possible since the splitting error of PISO is negligibly small at the level of time-step required to eliminate the temporal truncation error. A backward temporal difference scheme is used, while the convective terms are discretized using a second-order upwind difference scheme. The method can also be used for steady-state flows, e.g., Hadid *et al.* [17].

Calculations are performed for the turbulent flow around a square cylinder (step height, $H = 20$ mm) in a domain extending about $16 H$ downstream and $2.5 H$ upstream of the obstacle.

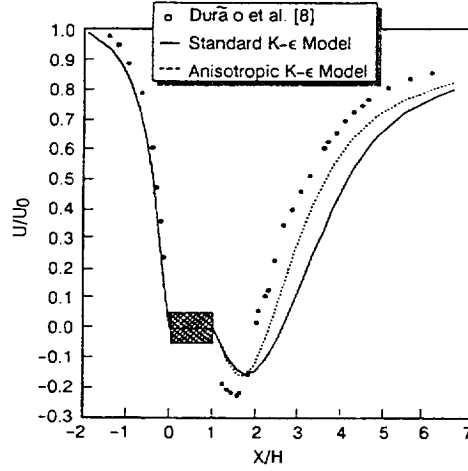


Fig.2: Centerline distribution of mean axial velocity

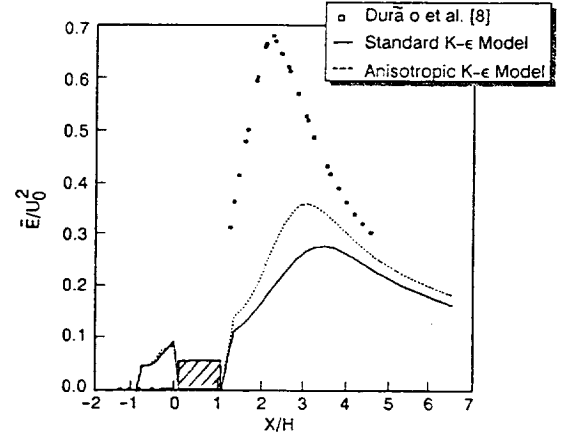


Fig.3: Time-mean kinetic energy of the velocity fluctuations

The calculations captured the vortex shedding phenomenon after perturbing the flow at the inlet. A reference velocity of 0.68 m/s and turbulence intensity of 6% (i.e., $k = \langle u'^2 \rangle = 3.6 \times 10^{-3} \text{ m}^2/\text{s}^2$) were used as the inlet conditions. The length scale L of turbulence at the inlet was not measured in the experiment but an order of $L \sim 0.1 \text{ mm}$ was assumed from which the energy dissipation rate $\varepsilon = k^{3/2}/L$ was estimated. It is expected that the calculated results are not sensitive to the precise value of ε used at the inlet. The upper and lower boundaries are treated as symmetry planes, at the exit, a zero-gradient outflow boundary condition is applied to each variable. The computational domain is resolved by 75×40 grid cells with clustering at the obstacle walls. An optimized time step of 0.001 sec. was chosen for the calculations.

4. RESULTS AND DISCUSSION

Figure 1(a) shows the normal velocity history at the centerline of the wake for $Re = 14000$ at five step heights downstream. The power spectrum of the normal velocity fluctuations (Fig.1(b)) confirms the oscillatory nature of the flow with a single predominant frequency of about 4.7 Hz, which is in agreement with experimental results [8]. Figure 1(c) shows a marker particle trace at time=3 sec., which illustrates the shedding pattern. In order to calculate the time-mean kinetic energy of the velocity fluctuations, the fluctuating velocity component (organized + turbulent) is $\hat{u}_i = u_i - \bar{U}_i$. For the 2-D plane geometry considered, the kinetic energy of the velocity fluctuations can be written as;

$$E = \frac{3}{4} (\hat{u}_1^2 + \hat{u}_2^2) \quad (12)$$

where $\hat{u}_i^2 = u_i^2 - 2u_i\bar{U}_i + \bar{U}_i^2$, and the time-mean value of the kinetic energy of the velocity fluctuations is

$$\bar{E} = \frac{3}{4} (\overline{\hat{u}_1^2} + \overline{\hat{u}_2^2}) \quad (13)$$

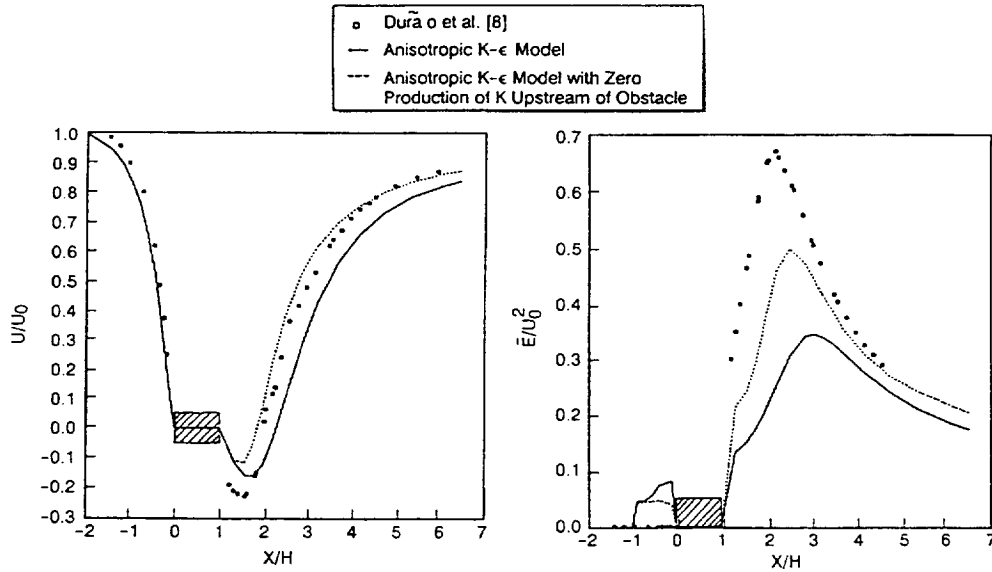


Fig.4: Centerline distribution of mean axial velocity

Fig.5: Time-mean kinetic energy of the velocity fluctuations

where

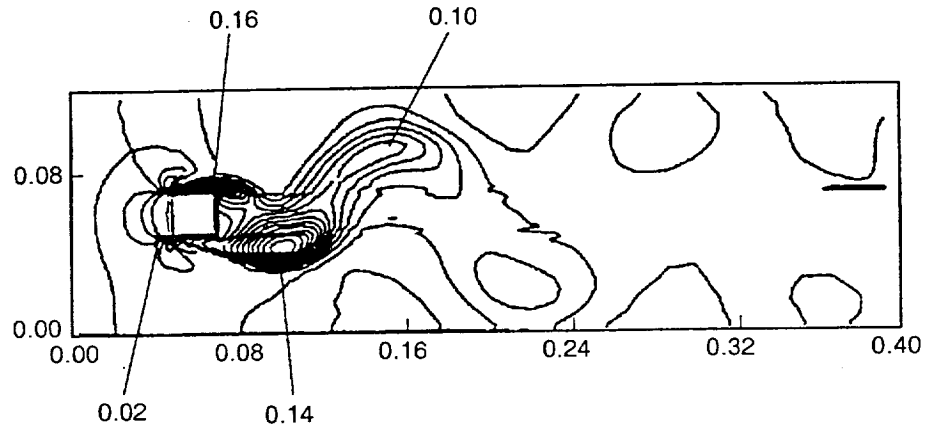
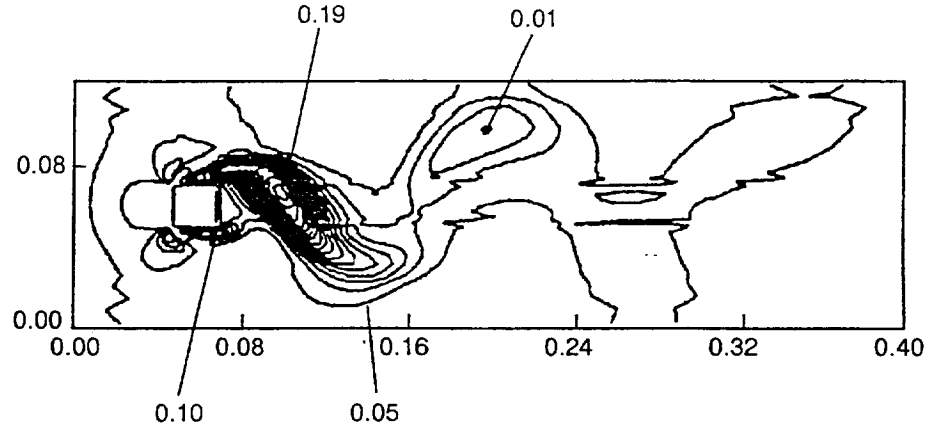
$$\overline{u_i^2} = (\overline{u_i^2} - 2\overline{u_i U_i} + \overline{U_i^2}) = \overline{[(U_i + u'_i)^2 - 2(U_i + u'_i)U_i + U_i^2]}$$

and from the definition of time averaging $\overline{u'_i U_i} = 0$, we get,

$$\overline{u_i^2} = \overline{U_i^2} - \overline{U_i^2} + \overline{u_i'^2} \quad (i = 1, 2) \quad (14)$$

The first two terms on the right hand side of (14) represent the organized periodic energy contribution, while the last term represents the turbulent energy contribution.

Figure 2 shows the distribution of the mean axial velocity at the centerline. The anisotropic model gives better distribution downstream of the obstacle. Figure 3 compares the calculated distribution of the time-mean kinetic energy of the fluctuating motion (periodic + turbulent) along the centerline of the flow. The figure shows a better trend exhibited by the anisotropic k- ϵ model due to the improved resolution of the normal stresses. The standard k- ϵ model acts to damp the periodic fluctuations by producing too much eddy viscosity, which underestimates the time-averaged momentum transfer. Hence, the length of the separation region behind the obstacle is overpredicted. Also, the maximum of the kinetic energy at the centerline lies further downstream. The length of the recirculation zone and the location of the maximum fluctuating energy are improved by using the anisotropic model. The figure also shows some fluctuating energy in front of the obstacle, whereas measurements indicated that the flow remained virtually laminar there. This is because in the k- ϵ model the large velocity gradients at the stagnation region produce large turbulent kinetic energy. Results are also obtained from calculations in which the production of k in front of the obstacle was suppressed. Figure 4 shows the mean axial velocity distribution indicating better comparison with the experiment downstream of the obstacle. Figure 5 shows the distribution of the mean kinetic energy along the centerline. It can be seen that suppressing the production of the kinetic energy in front of the obstacle causes an

(a) Using the standard $k-\epsilon$ model(b) Using the anisotropic $k-\epsilon$ modelFig.6: $\langle v'v' \rangle / U^2$ contours at $T = 3$ sec

increase in the fluctuating energy. Also, the peak of the energy fluctuations is shifted slightly upstream closer to the experimental data. The figure also shows smaller residual fluctuating energy in front of the obstacle. Figure 6(a) and (b) show the contour plots of the normal turbulent stress term $\langle v'v' \rangle$ at an instant $T = 3$ sec. It can be seen that the anisotropic $k-\epsilon$ model produces higher $\langle v'v' \rangle$ values, which act to increase the total fluctuating energy.

5. CONCLUSIONS

The turbulent vortex shedding flow behind a square cylinder was analyzed using an efficient time-accurate numerical method based on the PISO methodology. Turbulence was modeled using an anisotropic $k-\epsilon$ model which resolves the anisotropy of the Reynolds stresses reasonably well. Comparisons with the experimental data show the advantages of the model as compared with the standard isotropic $k-\epsilon$ model. Accurate predictions, however, can only be made by accounting for the history and transport effects of the individual Reynolds stresses. The anisotropic $k-\epsilon$ model seems to offer a compromise between the computationally intensive

Reynolds stress model and the standard isotropic $k-\epsilon$ model.

REFERENCES

- [1] R.Davis and E.Moore (1982) A numerical study of vortex shedding from rectangles; *J. Fluid Mech.* vol.116 p.475
- [2] M.Braza, P.Chassaing and H.Ha Minh (1986) Numerical study and physical analysis of the pressure and velocity fields in the near wake of a cylinder; *J. Fluid Mech.* vol.165 p.79
- [3] R.Franke, W.Rodi and B.Schönung (1989) Analysis of experimental vortex shedding data with respect to turbulence modelling; 7th Symp. on Turbulent Shear Flows, Stanford Univ., Aug. 21, 1989
- [4] B.Cantwell and D.Coles (1983) An experimental study of entrainment and transport in the turbulent near wake of a circular cylinder; *J. Fluid Mech.* vol.136 p.321
- [5] J.MacInnes R.Claus and P.Huang (1989) Time-dependent calculation of a forced mixing layer using a $k-\epsilon$ turbulence model; 7th Symp. on Turbulent Shear Flows, Stanford Univ., Aug. 21, 1989
- [6] I.Weisbrot and I. Wygnanski (1989) On coherent structures in a highly excited mixing layer; *J. Fluid Mech.*, vol.195 p.321
- [7] S.Majumdar and W.Rodi (1985) Numerical calculations of turbulent flow past circular cylinder; 3rd Symp. on Numerical and Physical Aspects of Aerodynamic Flows, Long Beach, California, 1985
- [8] D.Durão, M.Heitor and J.Pereira (1988) Measurements of turbulent and periodic flows around a square cross-section cylinder; *Experiments in Fluids*, vol.6 p.298
- [9] R.Issa (1986) Solution of the implicitly discretized fluid flow equations by operator-splitting; *J. Comput. Phys.*, vol.62 p.40
- [10] A.Yoshizawa (1984) Statistical analysis of the deviation of the Reynolds stress from its eddy-viscosity representation; *Phys. Fluids*, vol.27 p.1377
- [11] C.Speziale (1987) On nonlinear $k-l$ and $k-\epsilon$ models of turbulence; *J. Fluid Mech.*, vol.178 p.459
- [12] B.Launder and D.Spalding (1974) The numerical computation of turbulent flows; *Comput. Meth. Appl. Mech. Engng.*, vol.3 p.269
- [13] S.Nisizima and A.Yoshizawa (1987) Turbulent channel and Couette flows using an anisotropic $k-\epsilon$ model; *AIAA J.*, vol.25 p.414
- [14] H.Myong and N.Kasagi (1990) Prediction of anisotropy of the near-wall turbulence with an anisotropic low-Reynolds number $k-\epsilon$ turbulence model; *J. Fluid Engng.*, vol.112 p.521
- [15] W.Jones and B.Launder (1973) The calculation of low-Reynolds number phenomena with a two-equation model of turbulence; *Int. J. Heat Mass Transfer*, vol.16 p.1119
- [16] H.Chen and V.Patel (1988) Near-wall turbulence models for complex flows including separation; *AIAA J.*, vol.26 p.641
- [17] A.Hadid, D.Chan, R.Issa and M.Sindir (1988) Convergence and accuracy of pressure-based finite difference schemes for incompressible viscous flow calculations in nonorthogonal coordinate system; *AIAA Paper* 88-3529

Single point modeling of rotating turbulent flows

By A. H. Hadid¹, N. N. Mansour² AND O. Zeman³

A model for the effects of rotation on turbulence is proposed and tested. These effects which influence mainly the rate of turbulence decay are modeled in a modified turbulent energy dissipation rate equation that has explicit dependence on the mean rotation rate. An appropriate definition of the rotation rate derived from critical point theory and based on the invariants of the deformation tensor is proposed. The modeled dissipation rate equation is numerically well behaved and can be used in conjunction with any level of turbulence closure. The model is applied to the two-equation k - ϵ turbulence model and is used to compute separated flows in a backward-facing step and an axisymmetric swirling coaxial jets into a sudden expansion. In general, the rotation modified dissipation rate model show some improvements over the standard k - ϵ model.

1. Motivation and objectives

The ability to accurately model the effects of rotation on turbulence has a wide variety of important applications in rotating machinery and combustion devices. Many turbulent flows of engineering importance involve combinations of rotational and irrotational strains. However, turbulence models of the eddy viscosity type are oblivious to the presence of rotational strains since they depend only on the mean velocity gradients through their symmetric part (i.e. the mean rate of strain tensor). The rotation rate, for example, does not explicitly enter the equations for the turbulent kinetic energy and its dissipation rate, yet evidence from experiments (Wigeland and Nagib 1978, Jacquin *et al.* 1990) and from direct numerical simulation (Bardina *et al.* 1985, Speziale *et al.* 1987, Mansour *et al.* 1991) show that the decay rate of turbulence is reduced by the presence of rotation.

The effects of rotation on turbulence are known to be subtle. They are manifested through changes in the spectrum of the turbulence caused by nonlinear interactions. For initially isotropic turbulence, rotation inhibits the cascade of energy from large to small scales. Zeman (1994) proposed a modified energy spectrum that takes into account the effects of rotation at high Reynolds number by introducing a rotation wavenumber, $k_\Omega = \sqrt{\Omega^3/\epsilon}$, below which rotation effects on spectral transfer are important. Much of the application work in simulating rotating flows have been conducted using varieties of eddy viscosity models (k - ϵ or k - l) and second order closure models with modified dissipation rate transport equation to account for

1 Rocketdyne Division/Rockwell International

2 NASA/Ames Research Center

2 Center for Turbulence Research

rotational effects. However, most of these models fail to predict the asymptotic behavior of the turbulence decay rate in the limits of large rotation rate. The objectives of this work are to model the effects of rotation using single-point two equation models and to offer an appropriate definition of the mean rotation rate that is consistent with the fact that spin is the main cause of reduction in the dissipation rate.

2. Accomplishments

For incompressible viscous flow with constant properties, the modeled transport equations for the turbulent kinetic energy, k , and its dissipation rate, ϵ , that are widely used for engineering applications take the form;

$$k_{,t} + U_j k_{,j} = D_k + P_k - \epsilon \quad (1)$$

$$\epsilon_{,t} + U_j \epsilon_{,j} = D_\epsilon + P_\epsilon - \Phi_\epsilon \quad (2)$$

where D_k and D_ϵ are the diffusion terms for k and ϵ respectively and are modeled as

$$D_k = \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) k_{,j} \right]_{,j}, \quad D_\epsilon = \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \epsilon_{,j} \right]_{,j},$$

where ν is the laminar viscosity and ν_t is the eddy viscosity $= C_\mu k^2/\epsilon$. σ_k and σ_ϵ are the ratio of Prandtl to Schmidt numbers and are taken as constants. P_k is the production term for k given as $P_k = -\overline{u'_i u'_j} U_{i,j}$, where $\overline{u'_i u'_j}$ is the Reynolds stress term and U_i is the mean velocity in the i -direction.

Assuming that the production of the dissipation rate P_ϵ is proportional to the production of turbulent kinetic energy P_k , i.e. $P_\epsilon \sim P_k/T$ where T is the turbulent time scale given by $T = k/\epsilon$. Similarly assume that the destruction rate of dissipation rate Φ_ϵ is proportional to the turbulent energy dissipation rate term, i.e. $\Phi_\epsilon \sim \epsilon/T$. The modeled form of the dissipation rate equation becomes

$$\epsilon_{,t} + U_j \epsilon_{,j} = D_\epsilon + C_1 \frac{\epsilon}{k} P_k - C_2 \frac{\epsilon^2}{k} \quad (3)$$

Due to the symmetry of the Reynolds stress tensor $\overline{u'_i u'_j}$, the kinetic energy production term can be written as $P_k = -\overline{u'_i u'_j} S_{ij}$, where $S_{ij} = (U_{i,j} + U_{j,i})/2$ is the mean rate of strain tensor. Therefor it can be seen that the standard dissipation rate, eq. (3), has no explicit dependence on the mean rotation tensor $\Omega_{ij} = (U_{i,j} - U_{j,i})/2$. It follows that the commonly used modeled dissipation rate equation can only be affected indirectly by rotational strains through the changes that they induce in the Reynolds stress tensor.

In order to sensitize the dissipation rate equation to rotational effects, consider the simple case of isotropic turbulence in a rotating frame. In this case, an initially decaying isotropic turbulence is described by;

$$k_{,t} = -\epsilon \quad (4)$$

$$\epsilon_{,t} = -C_2 \frac{\epsilon^2}{k} \quad (5)$$

Equations (4) and (5) do not distinguish the difference between isotropic turbulence in a rotating frame and in an inertial frame. Models that have a non zero rotational correction have been proposed by Bardina *et al.* (1985), for example, for rotating isotropic turbulence where eq. (5) takes the form

$$\epsilon_{,t} = -C_2 \frac{\epsilon^2}{k} - C_3 \Omega \epsilon \quad (6)$$

with $C_2 = 1.83$ and $C_3 = 0.15$.

The above model is able only to accurately predict the reduction in the decay rate of the turbulent kinetic energy in rotating isotropic turbulence for weak to moderate rotation rates where the effects are small. However, for sufficiently high rotation rates and long enough time, the model drastically underpredicts the decay rate of the turbulent kinetic energy.

Hanjalic and Launder (1980) proposed a model for which the ϵ -transport equation in rotating isotropic turbulence takes the form

$$\epsilon_{,t} = -C_2 \frac{\epsilon^2}{k} - C_3 \Omega^2 k \quad (7)$$

where $C_2 = 1.92$ and $C_3 = 0.27$.

This model predicts unphysical behavior of negative dissipation rate at high rotation rates, thus violating the realizability constraint. Other modifications to the dissipation rate transport equation have been proposed to account for rotational strains, e.g Raj (1975) and Pope (1978). Again they fail in one way or another to account accurately for the rotational effects.

3. Proposed model

In the present work a new model is proposed that accounts for rotational effects and correctly predicts the asymptotic behavior at zero to infinite rotation rates. Consider the dissipation rate equation in rotating isotropic turbulence

$$\epsilon_{,t} = - \left(1.7 + \frac{5}{6} \frac{\alpha^2}{\alpha^2 + 1} \right) \frac{\epsilon^2}{k} \quad (8)$$

with

$$\alpha = 0.35 Ro^{-1} \quad (9)$$

where Ro is the Rossby number defined as $Ro^{-1} = \Omega k / \epsilon$. For $\Omega \gg 1$, $C_2 = 2.5$, which gives a power law exponent $n = 0.6$ (in $k \sim t^{-n}$) matching the power law proposed by Squires *et al.* (1993) for the asymptotic state of rotating homogeneous turbulence at high Reynolds numbers.

The experimental data of Jacquin *et al.* (1990) are used to test the proposed model. Their experiments consisted of measuring the velocity field and characteristic quantities characterizing the fluctuating field downstream of a rotating cylinder

containing a honeycomb structure and a turbulence producing grid. The coupled differential equations for k and ϵ describing the effects of rotation on an initially isotropic turbulence can be written as

$$k_t = -\epsilon \quad (10)$$

$$\epsilon_t = - \left(C_2 + C_3 \frac{\alpha^2}{\alpha^2 + 1} \right) \frac{\epsilon^2}{k} \quad (11)$$

These equations were solved numerically using a fourth-order Runge-Kutta integration scheme. The model predictions (with $C_2 = 1.7$ and $C_3 = 5/6$) are compared with the experimental data of Jacquin *et al.* (1990) as shown in Fig. 1a. The model predicts well the evolution of turbulent kinetic energy and its decay rate for a wide range of rotation rates. We have also tested the model for the three Reynolds numbers measured by Jacquin *et al.* (1990), and found similar agreement of the model predictions with the data. We should point out at this point that the value $C_2 = 1.7$, proposed here for zero rotation rate, is lower than the value conventionally used in k - ϵ modeling. We find that with the conventional value of $C_2 = 1.92$ (and $C_3 = 3/5$) the model fails to predict the experimental data (see Fig. 1b)

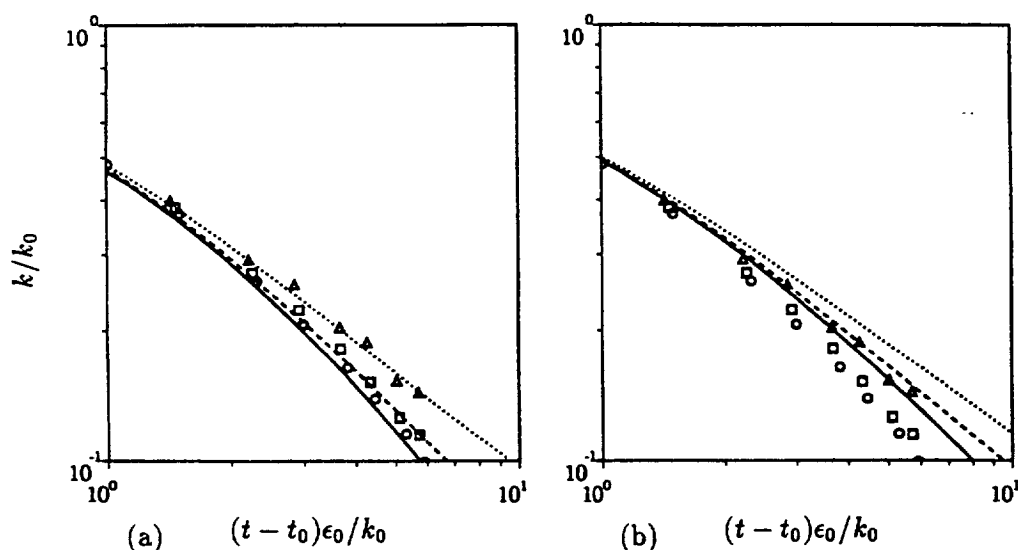


FIGURE 1. Decay of turbulent kinetic energy. Symbols are the data of Jacquin *et al.* (1990), lines are the model predictions. \circ & — $\Omega = 62.8$ (rad/s), \square & - - - $\Omega = 31.4$ (rad/s), \triangle & $\Omega = 15.7$. (a) Model predictions with $C_2 = 1.7$ and $C_3 = 5/6$; (b) Model predictions with $C_2 = 1.92$ and $C_3 = 3/5$.

4. Rotation Rate For General Flows

In order to test the rotational correction proposed in eq. (8) to the dissipation rate equation for general flows where the rotation rate is a function of position and

in the presence of mean strains, the question arises as to what is the appropriate definition of the rotation rate, Ω ?

In most previous studies, the rotation rate or the mean vorticity Ω was replaced by $\sqrt{\Omega_{ij}\Omega_{ij}}/2$, where $\Omega_{ij} = (U_{i,j} - U_{j,i})/2$ is the rotation rate tensor of the mean flow. However, such definition does not distinguish between a vortex sheet and a vortex. A definition of a vortex or a region of vorticity (with spin) was given by Chong *et al.* (1990) -using the arguments of the critical point theory and the invariants of the deformation tensor- as a region in space where the vorticity is sufficiently strong to cause the rate of strain tensor to be dominated by the rotation tensor, i.e. the rate of deformation tensor has complex eigenvalues. This definition satisfies the principle of frame invariance since it depends only on the properties of the deformation tensor. We shall use it because the reduction in the dissipation rate is due mainly to the spin that the mean imposes on the turbulence. Consider the matrix D_{ij} of the elements of the deformation tensor,

$$D_{ij} = U_{i,j} \quad (12)$$

which can be split to

$$D_{ij} = S_{ij} + \Omega_{ij} \quad (13)$$

The complex eigenvalues of D_{ij} are found by solving the characteristic equation $|D_{ij} - \lambda\delta_{ij}| = 0$, where the λ 's are the eigenvalues of D_{ij} . For a 3×3 matrix, λ can be found from the solution of

$$\lambda^3 + P\lambda^2 + Q\lambda + R = 0 \quad (14)$$

where P , Q and R are the matrix invariants and are given by

$$P = -U_{i,i} \quad (15)$$

$$Q = \frac{1}{2}(P^2 - S_{ij}S_{ji} - \Omega_{ij}\Omega_{ji}) \quad (16)$$

$$R = \frac{1}{3}(-P^3 + 3PQ - S_{ij}S_{jk}S_{ki} - 3\Omega_{ij}\Omega_{jk}S_{ki}) \quad (17)$$

For an incompressible flow $P = 0$ from continuity and the characteristic equation becomes

$$\lambda^3 + Q\lambda + R = 0 \quad (18)$$

Now if

$$A = \left[-\frac{R}{2} + \sqrt{\left(\frac{R^2}{4} + \frac{Q^3}{27}\right)} \right]^{1/3}$$

and,

$$B = \left[-\frac{R}{2} - \sqrt{\left(\frac{R^2}{4} + \frac{Q^3}{27}\right)} \right]^{1/3}$$

then the three roots of λ are;

$$\left[A + B, -\frac{A+B}{2} + i\frac{A-B}{2}\sqrt{3}, -\frac{A+B}{2} - i\frac{A-B}{2}\sqrt{3} \right]$$

That is λ can have:

- (i) all real roots which are distinct when

$$[(Q/3)^3 + (R/2)^2] < 0,$$

or

- (ii) all real roots where at least two roots are equal when

$$[(Q/3)^3 + (R/2)^2] = 0,$$

or

- (iii) one real root and a pair of complex conjugate roots when

$$[(Q/3)^3 + (R/2)^2] > 0.$$

We shall follow Chong *et al.* (1990) and define the rotation rate

$$\Omega = \Im(\lambda) = \frac{\sqrt{3}}{2}(A - B), \quad \text{when } [(Q/3)^3 + (R/2)^2] > 0, \quad (19)$$

$\Omega = 0$ otherwise. It is important to note that for two dimensional Cartesian flows, the rotation rate defined by Eq. (19) reduces to $\Omega = \sqrt{|Q|}$, when Q , the determinant of the deformation tensor matrix, is negative. For pure shear the definition, eq. (19) yields $\Omega = 0$. Conventional models that are calibrated for shear flows, need not be recalibrated when corrections based on Ω are added to the model.

5. Numerical Procedure

For a two-dimensional, incompressible and steady turbulent flow, the Reynolds averaged momentum, continuity, turbulent kinetic energy and dissipation rate equations can be written in the generalized form;

$$\frac{\partial}{\partial x}(\rho U \Phi) + \frac{1}{r} \frac{\partial}{\partial y}(\rho r V \Phi) = \frac{\partial}{\partial x} \left(\Gamma_{\Phi} \frac{\partial \Phi}{\partial x} \right) + \frac{1}{r} \frac{\partial}{\partial y} \left(r \Gamma_{\Phi} \frac{\partial \Phi}{\partial y} \right) + S_{\Phi} \quad (20)$$

Where $r = 1$ for Cartesian two-dimensional flow, and $y = r$ for two-dimensional axisymmetric flow. Table 1 gives a summary of the terms in eq. (20) for the dependent variables solved in the code.

Φ	Γ_{Φ_z}	Γ_{Φ_r}	S_Φ
1	0.	0.	0.
U	$2\mu_e$	μ_e	$-\partial P/\partial x + 1/r \partial(\mu_e r \partial V/\partial x)/\partial y$
V	μ_e	$2\mu_e$	$-\partial P/\partial y + \partial(\mu_e \partial U/\partial y)/\partial y$
W	μ_e	μ_e	$-\rho VW/r - W/r^2 \partial(r\mu_e)/\partial r$
k	$\mu + \mu_t/\sigma_k$	$\mu + \mu_t/\sigma_k$	$P_k - \rho\epsilon$
ϵ	$\mu + \mu_t/\sigma_\epsilon$	$\mu + \mu_t/\sigma_k$	$C_1 P_k \epsilon/k - C_2 \rho \epsilon^2/k$

Table 1. Summary of the governing equations. ρ is the density, Γ_{Φ_z} and Γ_{Φ_r} are the exchange coefficients in the axial and radial directions respectively, S_Φ is the source term for the variable Φ . In the table, μ_e is the effective viscosity given as $\mu_e = \mu + \mu_t$, where μ is the laminar viscosity and μ_t is the turbulent viscosity, $\mu_t = C_\mu \rho k^2/\epsilon$.

In the standard k - ϵ turbulence model the constants C_μ , C_1 , C_2 , σ_k and σ_ϵ have the values 0.09, 1.44, 1.92, 1.0 and 1.0 respectively.

In the rotation modified k - ϵ turbulence model, only C_2 takes the form given by eq. (11) i.e., $C_2 = 1.7 + (5/6)\alpha^2/(\alpha^2 + 1)$.

The governing transport eq. (20) is solved using the primitive variables on a nonstaggered mesh and converted into a system of algebraic equations by integrating over control volumes defined around each grid point. The SIMPLE pressure-correction scheme (Patankar 1980) is used to couple the pressure and velocities and the resulting algebraic equations are solved iteratively. The convective terms are differenced using a second-order upwind scheme while the diffusion terms are approximated by a central differencing scheme. The physical domain is discretized using a non-uniform mesh where grid points are clustered close to the walls.

6. Model Application

The performance of the present model for complicated recirculating flows is demonstrated through calculations and comparisons with the experimental data of Driver & Seegmiller (1985) for backward-facing step flows and with the experiments of Roback & Johnson (1983) for a confined swirling coaxial jets into a sudden expansion.

Figure 2, shows the streamlines for the backward-facing step using the rotation modified k - ϵ turbulence model. The calculations were performed on a 100×40 grid points. The computational domain had a length of $50H$ (H is the step height) and a width of $9H$. The experimental data were used to specify the inflow conditions for a channel flow calculation where the fully developed profiles at the channel exit were used as the inlet conditions for the backward-facing step calculations. Fully developed flow conditions were imposed at the outflow boundary. The standard wall function approach (Launder & Spalding 1974) was used to bridge the viscous sublayer near the wall.

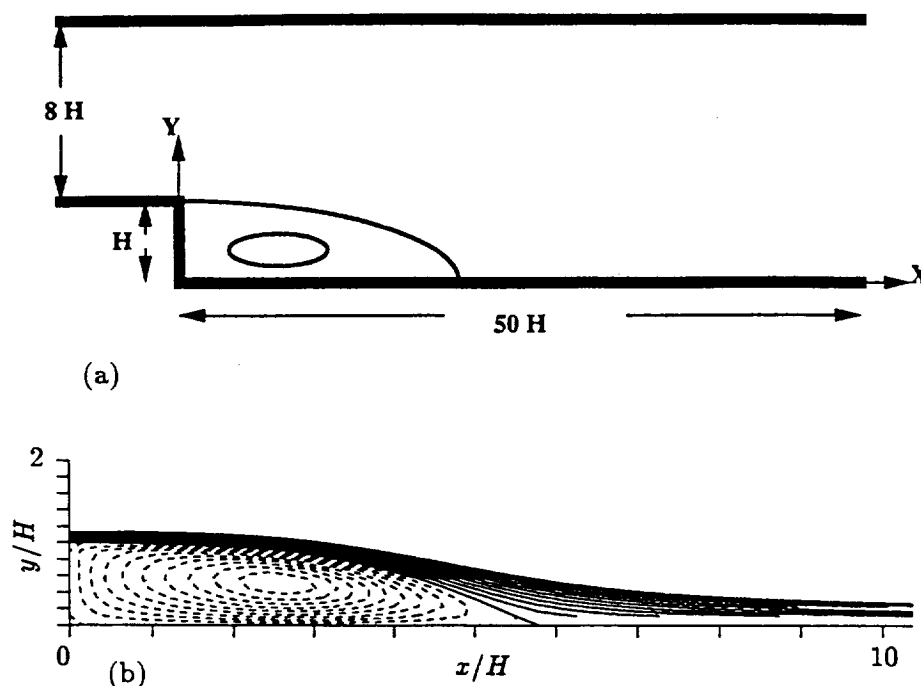


FIGURE 2. Backward-facing step geometry and stream-function contours. The contour levels were set between $(-0.1$ and $0.1)$ with an increment level $= 0.01$. ---- negative values, ——— positive values.

The computed reattachment lengths were $5.50H$ using the standard $k-\epsilon$ turbulence model and $6.22H$ for the rotation modified $k-\epsilon$ turbulence model. The modified $k-\epsilon$ model prediction is closer to the experimental value of $6.10H$. While these results are encouraging, they are mainly due to the fact that we have changed the value of C_2 for the non-rotating case. In general, a change in the value of C_2 will result in poor predictions of the mean profiles. The mean velocity profile at three locations downstream are shown on Fig. 3, while the turbulent stress profiles at $X/H = 4$ are shown on Fig. 4. All the quantities were normalized by the step height (H) and the experimental reference free-stream velocity (U_{ref}). It can be seen that the overall performance of the rotation modified dissipation rate equation is better than the standard $k-\epsilon$ model especially in the recirculation region (Figs. 3a, and 4). Some improvements are also obtained in the recovery region using the modified $k-\epsilon$ model. Figure 5 shows the contours of the effective rotation rate used as defined by Eq. (19).

For the 2D/axisymmetric swirling flow computations, the expressions for the invariants Q and R (Eqs. (16) & (17) respectively) are expanded and Eq. (19) is used to obtain the values of Ω . The model was used to predict the mean profiles for a confined double concentric jets with a swirling outer jet flow into a sudden expansion (Roback & Johnson, 1983, see Fig. 6). Measurements are available for the mean

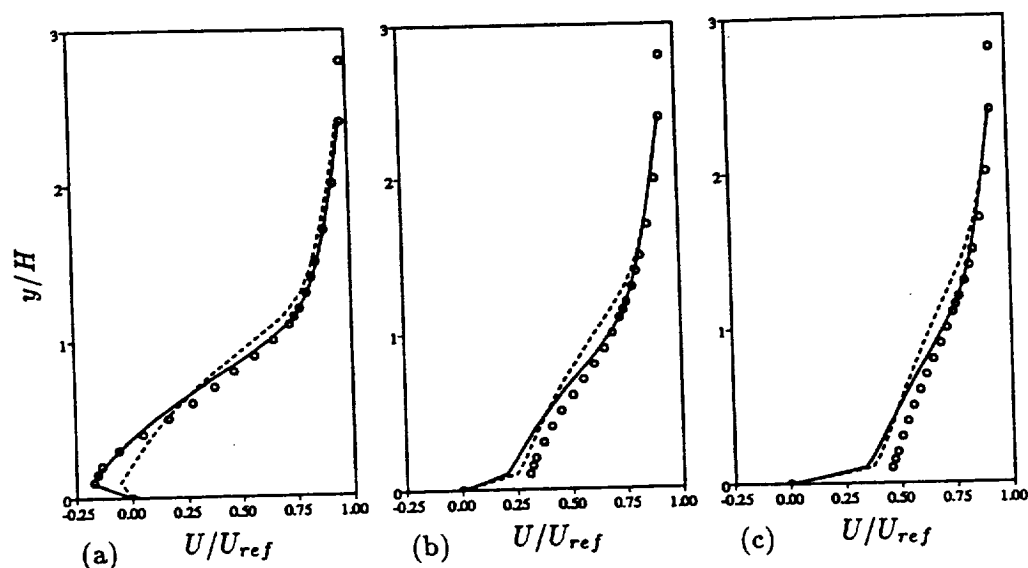


FIGURE 3. Mean axial velocity profiles at different axial locations. \circ data (Driver & Seegmiller, 1985); — modified $k-\epsilon$ model; ---- standard $k-\epsilon$ model. (a) $X/H = 4$, (b) $X/H = 8$, (c) $X/H = 12$.

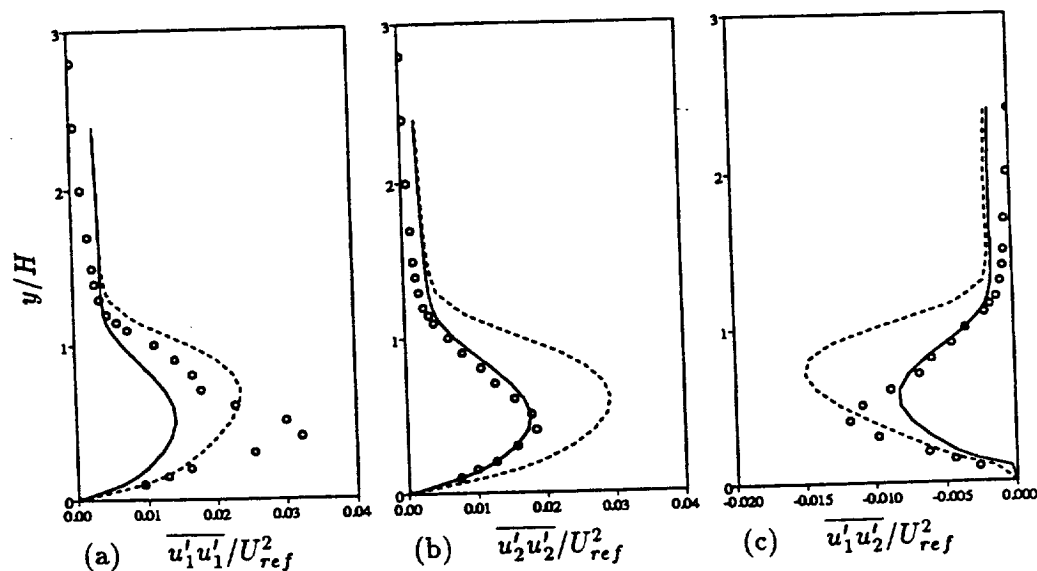


FIGURE 4. Turbulent stress profiles at $X/H = 4$. \circ data (Driver & Seegmiller, 1985); — modified $k-\epsilon$ model; ---- standard $k-\epsilon$ model. (a) $\overline{u'_1 u'_1} / U_{ref}^2$, (b) $\overline{u'_2 u'_2} / U_{ref}^2$, (c) $\overline{u'_1 u'_2} / U_{ref}^2$.

velocity profiles and velocity fluctuations downstream of the expansion. Simulations with a coarse nonuniform grid of 30×20 mesh points were made. However, there is some uncertainty about the inlet conditions to be used since the first velocity

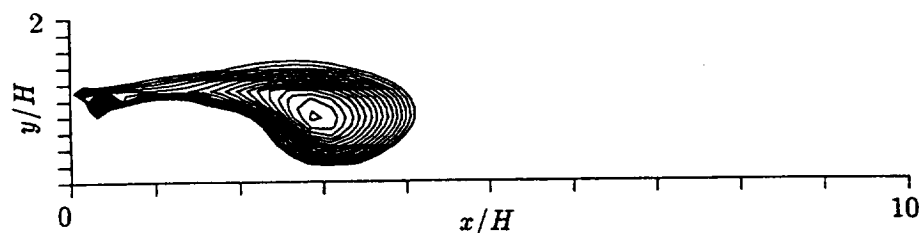


FIGURE 5. Contours of the effective rotation rate, Ω . Contour levels were set between (0.1,1.0) with an increment level = .01. are

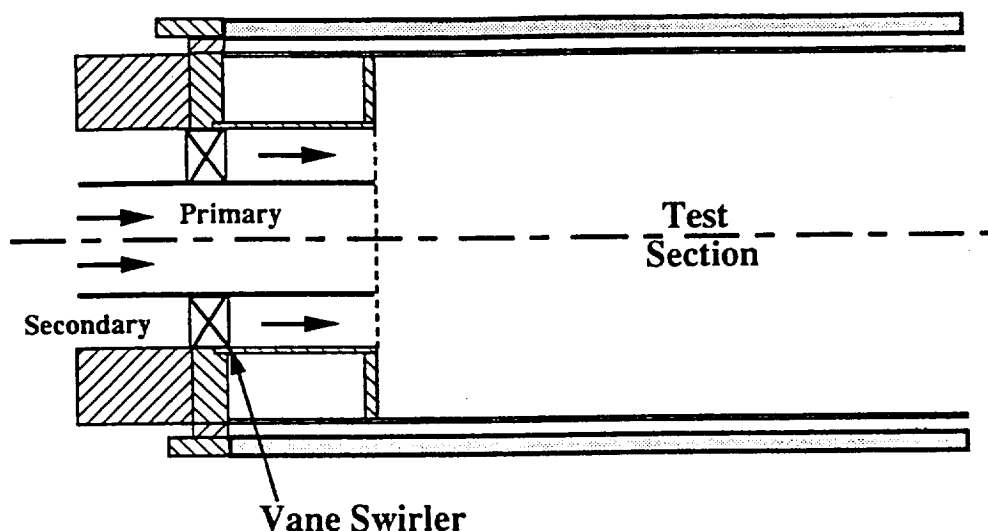


FIGURE 6. Roback & Johnson's swirling coaxial jets discharging into an expanded duct.

profiles measured were 5mm downstream of the expansion.

To predict this flow, the measured profiles at 5mm were adjusted near the edges and were used as inlet conditions at the expansion plane. Preliminary results obtained with the coarse mesh indicate similar trends as the experiment. Figure 7 shows the streamline contours using the standard and the modified $k-\epsilon$ turbulence models. The figure shows that a closed internal recirculation zone forms at the center with an additional zone at the corners downstream of the step. This causes a flow diversion outwards with high gradients between these regions. Figure 8 shows the axial and tangential velocity profiles at 25 mm downstream of the expansion using the standard and the modified $k-\epsilon$ turbulence models. Results in this case indicate little or no improvements offered using the modified $k-\epsilon$ model over the standard $k-\epsilon$ model. Finer mesh may improve the results but the uncertainties in the inlet boundary conditions raise the question about the adequacy of using this experiment for validation purposes.

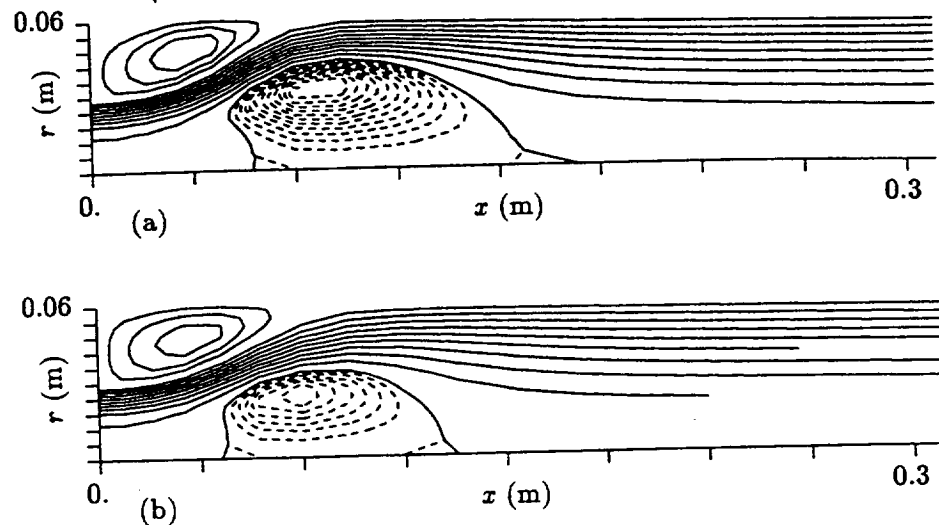


FIGURE 7. Swirling coaxial jets discharging into an expanded duct. Stream-function contour. ---- levels were set between $(-0.15, 0)$ with an increment level $= 0.01$, — levels were set between $(0, 0.7)$ with an increment level $= 0.05$. (a) Standard $k-\epsilon$ model, (b) Modified $k-\epsilon$ model.

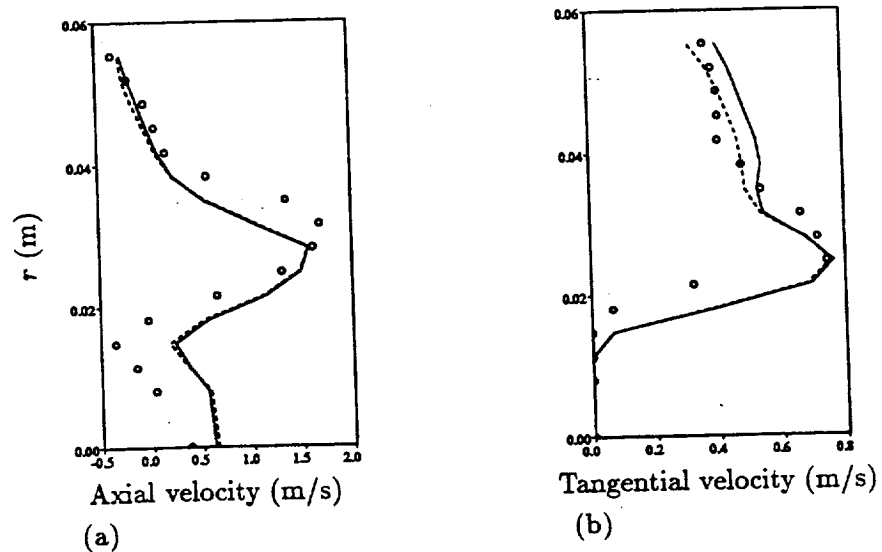


FIGURE 8. Velocity profiles at $X = 25$ mm. \circ data (Roback & Johnson, 1983); — modified $k-\epsilon$ model; ---- standard $k-\epsilon$ model. (a) Axial Velocity, (b) Tangential velocity.

7. Conclusions

A new simple model for the turbulent energy dissipation rate equation has been proposed to account for the rotational effects on turbulence. A frame invariant

definition of the rotation rate proposed by Chong *et al.* (1990) based on the critical point theory was used. The model can be used in conjunction with any level of turbulence closure. It was applied to the two-equation k - ϵ turbulence model and was tested for separated flows in a backward-facing step and for axisymmetric swirling jet into a sudden expansion. The model is numerically stable and showed improvements over the standard k - ϵ turbulence model. It is important to point out that the present study was carried out to roughly evaluate the model, but that a systematic recalibration of the constants in the k - ϵ model is needed before going any further with the proposed model.

The authors would like to acknowledge many discussions with Dr. K. Shariff regarding proper definition of the rotation rate.

REFERENCES

- BARDINA, J., FERZIGER, J. & ROGALLO, R. 1985 Effects of Rotation on Isotropic Turbulence: Computation and Modeling. *J. Fluid Mech.* **154**, 321-336.
- DRIVER, D. & SEEGLER, H. 1985 Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow. *AIAA Journal*. **23**, 163-171.
- HANJALIC, K. & LAUNDER, B. 1980 Sensitizing the Dissipation Equation to Irrational Strains. *ASME J. Fluids Eng.* **102**, 34-40.
- JACQUIN, L., LEUCHTER, O., CAMBON, C. & MATHIEU J. 1990 Homogenous Turbulence in the Presence of Rotation. *J. Fluid Mech.* **220**, 1-52.
- LAUNDER, B. & SPALDING, D. 1974 The Numerical Computation of Turbulent Flows. *Comput. Methods Appl. Mech. and Engg.* **3**, 269-289.
- MANSOUR, N. N., CAMBON, C. & SPEZIALE C. G. 1991 Theoretical and computational study of rotating isotropic turbulence. *Studies in Turbulence*, ed. by T. B. Gatski, S. Sarkar and C. G. Speziale, Springer Verlag, New-York.
- CHONG, M. S., PERRY, A. E. & CANTWELL, B. J. 1990 A General Classification of Three-dimensional Flow Fields. *Phys. Fluids A*. **2**, (5), 765-777.
- ROBACK, R. & JOHNSON, B. 1983 Mass and Momentum Turbulent Transport Experiment With Confined Swirling Co-Axial Jets. *NASA CR-168252*.
- SPEZIALE, C., MANSOUR, N. & ROGALLO, R. 1987 The Decay of Isotropic Turbulence in a Rapidly Rotating Frame. *Proceedings of the 1987 Summer Program*, CTR, NASA Ames/Stanford University.
- SQUIRES, K., CHASNOV, J., MANSOUR, N. & CAMBON, C. 1993 Investigation of the Asymptotic State of Rotating Turbulence Using LES. *Annual Research Briefs - 1990*, CTR, NASA Ames/Stanford University.
- WIGELAND, R. & NAGIB, H. 1978 Grid-Generated Turbulence With and Without Rotation About the Streamwise Direction. *IIT Fluids and Heat Transfer Report, R78-1*, Illinois Institute of Technology.
- ZEMAN, O. 1994 A Note on the Spectra and Decay of Rotating Homogeneous Turbulence. *Phys. Fluids*. **6**, 3221-3223.

1. Report No.		2. Government Accession No.		3. Recipient's Catalog No.	
Title and Subtitle Comapartive Study of Advanced Turbulence Models for Turbomachinery				5. Report Date October 1996	
7. Author(s) Ali H. Hadid and Munir M. Sindir				6. Performing Organization Code	
9. Performing Organization Name and Address Rocketdyne Division/Rockwell International 6633 Canoga Avenue, P. O. Box 7922 Canoga Park, CA 91303-7922				8. Performing Organization Report No.	
				10. Work Unit No.	
12. Sponsoring Agency Name and Address National Aeronautics and Space Adminstration George C. Marshall Space Flight Center Huntsville, AL 35812				11. Contract or Grant No. NAS8-38860	
				13. Type of Report and Period Covered Final Feb. 1992 - Oct. 1995	
15. Supplementary Notes Project Monitors, Dr. Paul McConnaughey and Mr. Robert Garcia, NASA/MSFC Program Manager, Dr. Glenn Havskjold, Rocketdyne Division/Rockwell International				14. sponsoring Agency Code	
16. Abstract <p>A computational study has been undertaken to study the performance of advanced phenomenological turbulence models coded in a modular form to describe incompressible turbulent flow behavior in two dimensional/axisymmetric and three dimensional complex geometry. The models include a variety of two equation models (single and multi-scale $k-\epsilon$ models with different near wall treatments) and second moment algebraic and full Reynolds stress closure models. These models were systematically assessed to evaluate their performance in complex flows with rotation, curvature and separation. The models are coded as self contained modules that can be interfaced with a number of flow solvers. These modules are stand alone satellite programs that come with their own formulation, finite-volume discretization scheme, solver and boundary condition implementation. They will take as input (from any generic Navier-Stokes solver) the velocity field, grid (structured H-type grid) and computational domain specification (boundary conditions), and will deliver, depending on the model used, turbulent viscosity, or the components of the Reynolds stress tensor. There are separate 2D/axisymmetric and/or 3D decks for each module considered.</p> <p>The modules are tested using Rocketdyn's proprietary code REACT. The code utilizes an efficient solution procedure to solve Navier-Stokes equations in a non-orthogonal body-fitted coordinate system. The differential equations are discretized over a finite-volume grid using a non-staggered variable arrangement and an efficient solution procedure based on the SIMPLE algorithm for the velocity-pressure coupling is used. The modules developed have been interfaced and tested using finite-volume, pressure-correction CFD solvers which are widely used in the CFD community. Other solvers can also be used to test these modules since they are independently structured with their own discretization scheme and solver methodology. Many of these modules have been independently tested by Professor C.P. Chen and his group at the University of Alabama at Huntsville (UAH) by interfacing them with own flow solver (MAST).</p>					
17. Key Words (Suggested by Author(s)) Turbulence Module decks, Swirling Flow, Turbulent Transport			18. Distribution Statement Unclassified - Unlimited		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of pages	
				22. Price*	